BUY IN PRINT WORLDWIDE MAGPI.CC/STORE

The MagPi

Issue 122 | October 2022 | magpi.cc

The official Raspberry Pi magazine

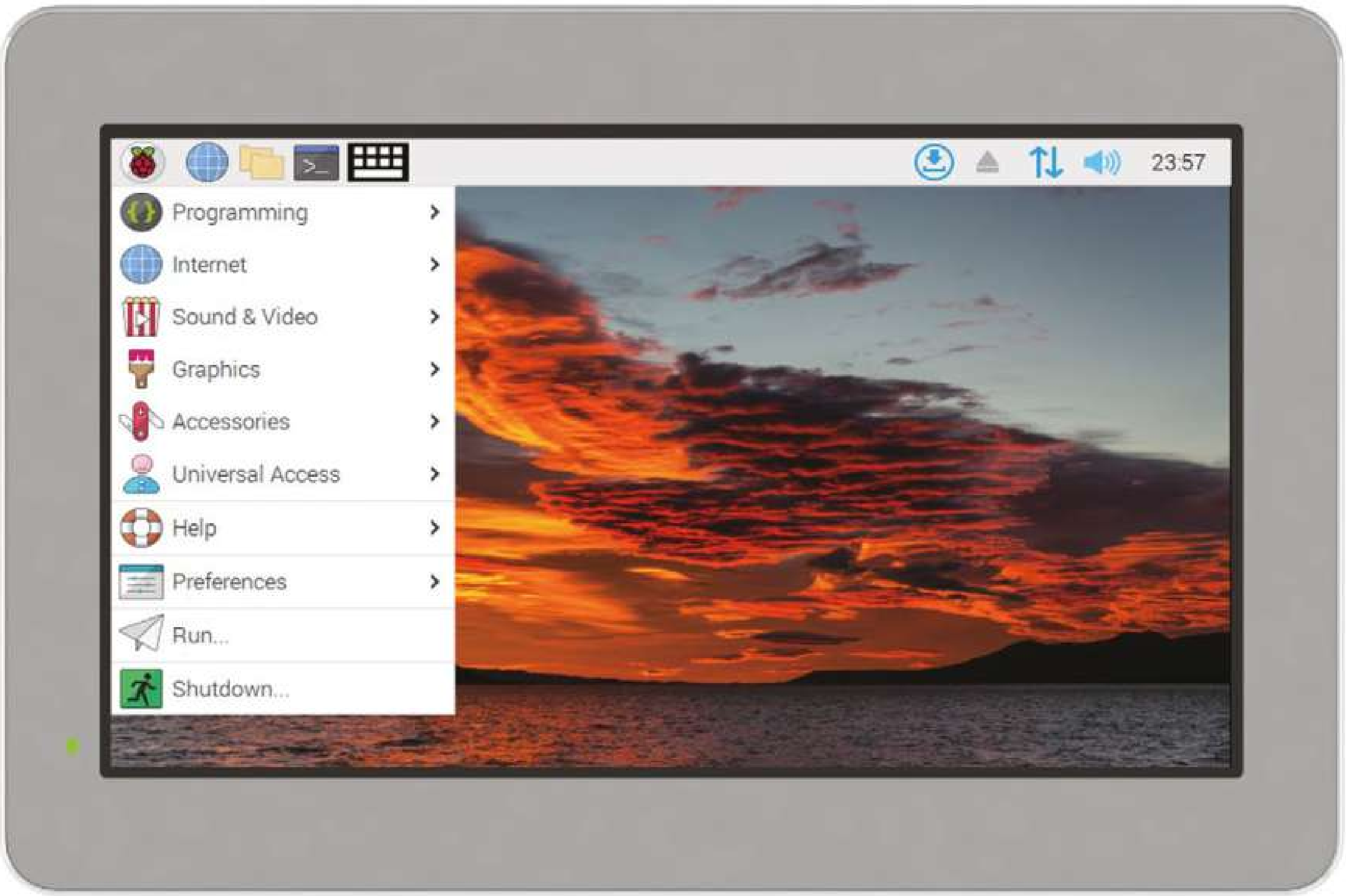
RETRO GAMING WITH PICO W

- > Classic Handhelds
- > Arcade Games
- > 1980s Computers

BUILD A MINI MAGIC MIRROR

MAKE A PICO MOOD LIGHT

THE TOP 10 RASPBERRY PI & PICO CASES


23:57
Programming
Internet
Sound & Video
Graphics
Accessories
Universal Access
Help
Preferences
Run...
Shutdown...

COMFILE

CP-IO22-A4-2
COMFILE
INPUT
OUTPUT
ADIN
DAOUT

WELCOME
to *The MagPi* 122

Retro gaming is part of Raspberry Pi that needs no introduction. We all grew up with video games and characters we know and love.

Creating a retro computer with Raspberry Pi is a rite of passage for new owners, and one of the best ways to rediscover the classics of years gone by.

Retro gaming with Raspberry Pi Pico, however, is a much newer concept. The RP2040-based microcontroller development board is increasingly capable of powering retro gaming and computer projects.

This month, we tasked KG Orphanides to put their considerable retro gaming expertise into squeezing the most out of Raspberry Pi Pico W for retro gaming (page 34).

Meanwhile, Rob has built a vacuum direct from Luigi's Mansion with Pico (page 58), and Phil has built a Mood Light with Raspberry Pi Pico (page 48).

There's a wealth of really great makes in this month's magazine. We hope it inspires you to start building your next project.

Lucy Hattersley Editor

EDITOR
Lucy Hattersley

Lucy is editor of *The MagPi* and just started a dancefloor on her own with no alcohol. Nerd superhero.

@LucyHattersley

GET A RASPBERRY PI PICO W WITH A SUBSCRIPTION! PAGE 28

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
RETRO GAMING WITH PICO W
Classic Handhelds
Arcade Games
1980s Computers
MAKE A PICO MOOD LIGHT
BUILD A MINI MAGIC MIRROR
THE TOP 10 RASPBERRY PI & PICO CASES

Contents

Issue 122 October 2022

Cover Feature

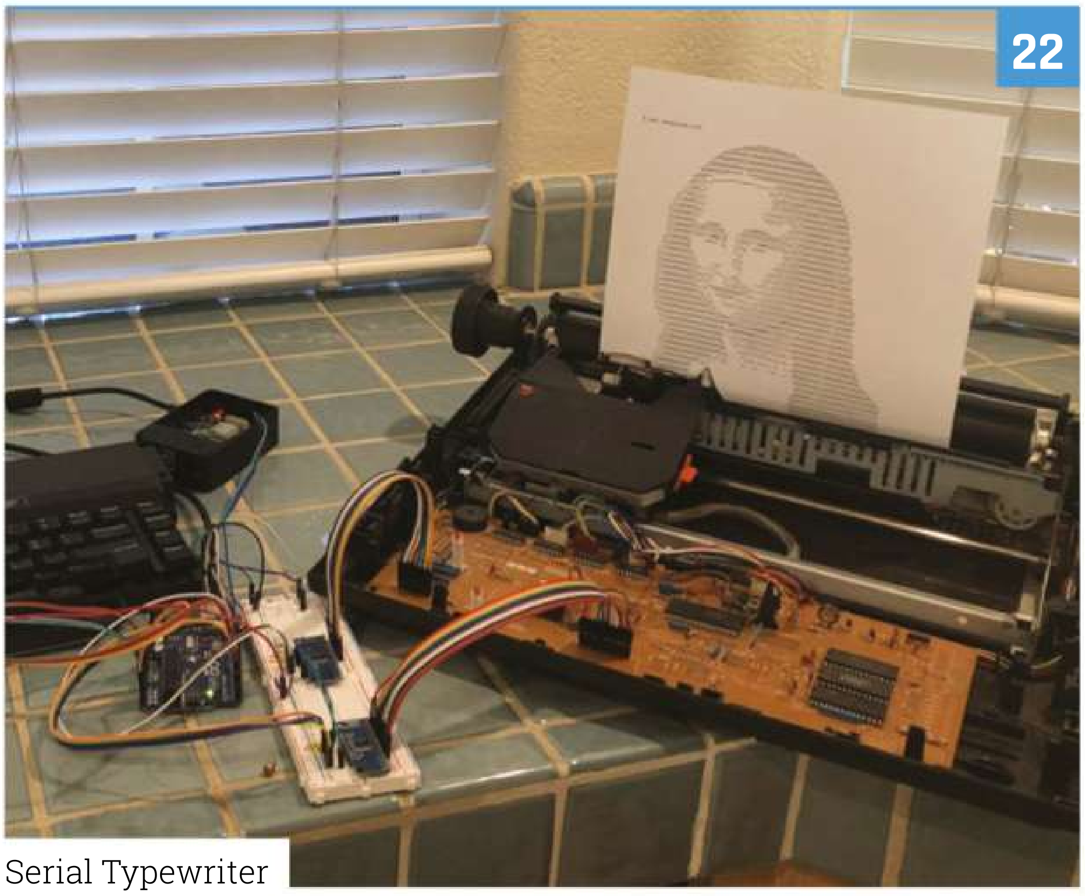
Regulars

Project Showcases

34

16

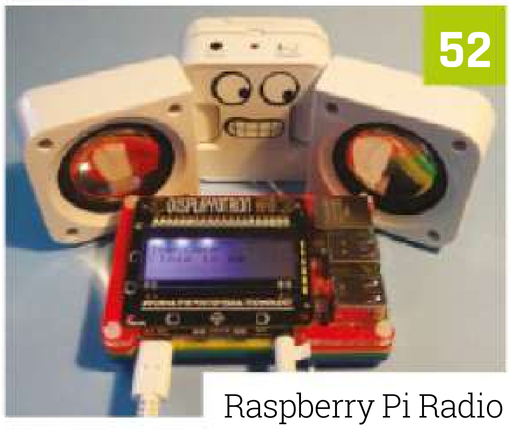
Pico Pocket Gamer


22

Serial Typewriter

The MagPi magazine is published monthly by Raspberry Pi Ltd. Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701, is the mailing agent for copies distributed in the US. Application to mail at Periodicals prices is pending at Williamsport, PA. POSTMASTER: Send address changes to The MagPi magazine c/o Publishers Service Associates, 2406 Reach Road, Williamsport, PA, 17701.

48
Mood Light


52
Raspberry Pi Radio

The Big Feature

68
Mini Magic Mirrors

77
Automation 2040 W

82
Centre for Computing History interview

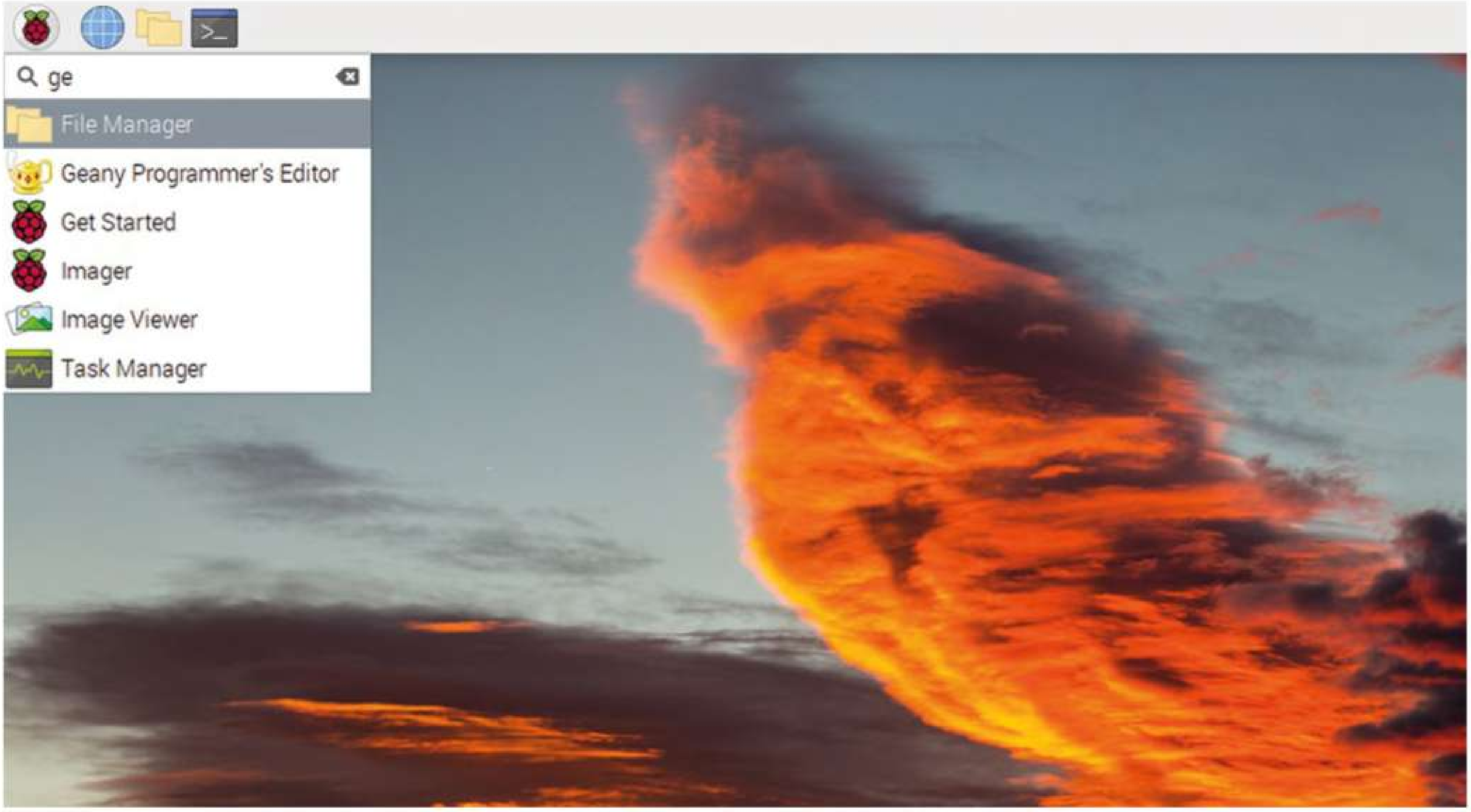
WIN 1 OF 5 EPD PICO KITS 95

DISCLAIMER: Some of the tools and techniques shown in The MagPi magazine are dangerous unless used with skill, experience, and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. Children should be supervised. Raspberry Pi Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in The MagPi magazine. Laws and regulations covering many of the topics in The MagPi magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in The MagPi magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

SELECTION
STARTS HERE
Digi-Key ELECTRONICS
DIGIKEY.CO.UK
cts
BROADCOM
intel
XP Power
TRACO POWER
SENSIRION
Sensata Technologies
SANYO DENKI
JST
BOURNS
RECOM
ublox
SILICON LABS
Littelfuse
onsemi
XILINX
Alpha Wire
KEMET
NXP
samtec
nichicon
VISHAY
TE
ANALOG DEVICES
Texas Instruments
Amphenol
molex
KYOCERA AVX
MICROCHIP
muRata
bel
Panasonic
OMRON
RENESAS
Micron
YAGEO
TAIYO YUDEN
VICOR
SAMSUNG
ALLEGRO
DELTA
Pulse
CREE LED
AAVID
Switchcraft
VERSALOGIC
Amulet
IQD
ODU
helix
LUXO
AMPLEON
THALES
antenova
SanKen

New update to Raspberry Pi OS

The latest version of the operating system introduces a search menu and new networking options. By **Simon Long**



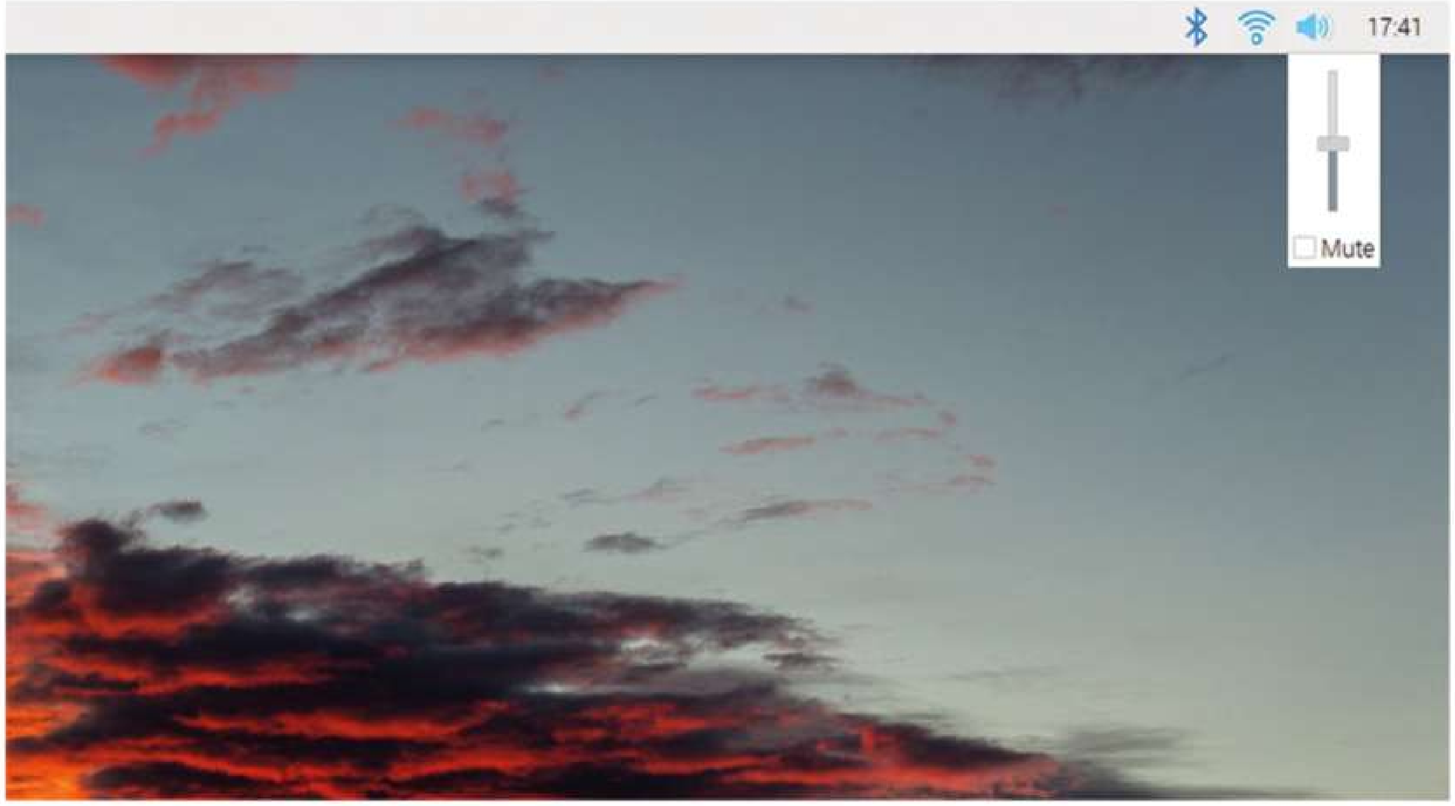
▲ Search items in the menu by pressing the Raspberry Pi (or Windows) key and entering text

Raspberry Pi has just released the latest version of Raspberry Pi OS. This time around, it is a wrapping-up of all the bug fixes and new versions of the software that have been released since the previous image in April. There are a few small tweaks to the user experience that are worth knowing.

New searchable main menu

For people who would rather type than move the mouse, Raspberry Pi has modified the main menu plug-in on the taskbar to allow text searching. Just hit the Raspberry key on your keyboard (which is usually the Windows key on a non-Raspberry Pi keyboard) to open the main menu, and start typing the name of the application you want to launch. When you start typing, a search box appears, along with a list of all the applications whose names contain the text you have typed.

You can move the cursor with the up and down arrow keys, and then just hit **ENTER** to launch the one you want, or double-click it with the mouse.



There is a new look for the audio control

If you don't start typing anything, the main menu will continue to work exactly as it did before; this is just additional functionality, not a replacement for the existing menu. Note that the search box only appears once you start to type text; you don't see it if you just open the menu and use it as normal.

New audio input control

In previous releases, the volume icon on the taskbar could be used to select both output and input audio devices. For this release, this has been split into two separate icons, one for output and one for input. If you connect an audio device which is capable of input, either via USB or Bluetooth, a microphone icon will appear on the taskbar next to the existing speaker icon.

To select the audio input device, right-clicking the microphone will show a list of the input devices available, and you can then click the one you want to use.

Also, left-clicking the microphone icon brings up a volume control which can be used to adjust the input level or mute the microphone completely.

Picamera2

This release includes the new Picamera2 Python camera interface. This is a higher-level interface than the existing libcamera and is easier to use.

New keyboard shortcuts

One thing that a few people have pointed out is that it is not possible to access the Bluetooth and wireless LAN menus on the taskbar from the keyboard; you need a mouse to use them.

In this release, Raspberry Pi has added keyboard shortcuts to access this functionality – hitting **CTRL+ALT+B** opens the Bluetooth menu, and **CTRL+ALT+W** opens the wireless LAN menu. Once the menu is open, the cursor up and down keys can be used to navigate. **ENTER** selects the highlighted item, and **ESCAPE** closes the menu.

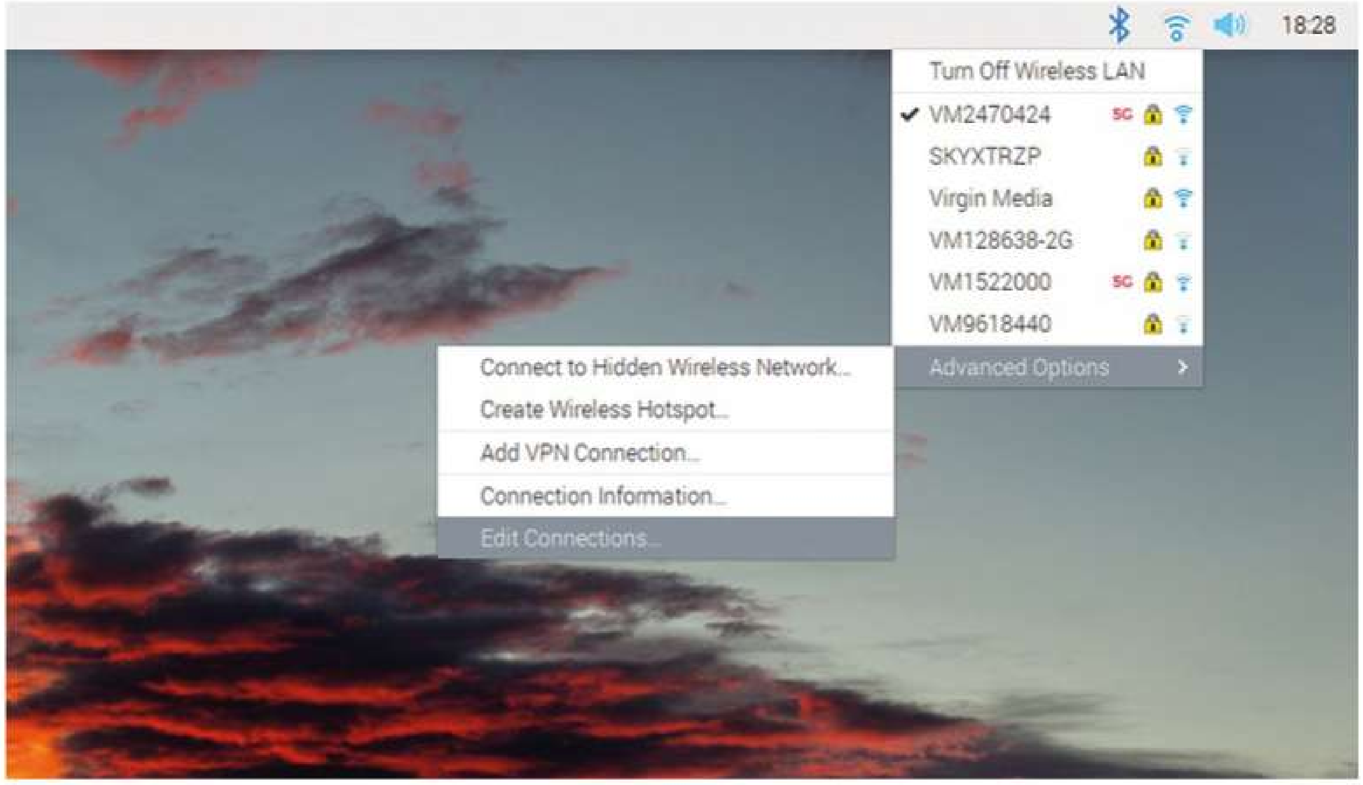
NetworkManager compatibility

In previous releases, Raspberry Pi has used a piece of software called dhcpcd to manage a lot of the networking functionality. In particular, it manages connection to WLAN networks – when you click the network icon on the taskbar and choose a network from the menu, this is all controlled by dhcpcd.

> Raspberry Pi has modified the main menu plug-in on the taskbar

Many other Linux distributions are now using a piece of software called NetworkManager to do the same job, which seems to be becoming the de facto standard, so they have added the option to use NetworkManager in Raspberry Pi OS. At present, dhcpcd is still the default – you need to switch to NetworkManager if you want to use it – but in some future release, NetworkManager will become the default.

NetworkManager adds a bunch of extra features which you may find useful. It allows you to easily connect to wireless networks with hidden SSIDs. It makes managing VPN connections more straightforward. And it allows you to easily configure a Raspberry Pi as a wireless access point.



▲ A new network manager is activated in raspi-config and offers additional features such as creating a wireless hotspot

NetworkManager support should be regarded as a beta feature for now – there may be a situation where something doesn't work as expected. If you rely on Raspberry Pi to have glitch-free networking, the safest thing to do is to stay on dhcpcd for now; but if you want the new features and are happy to live with the possibility that it may be slightly less reliable, then feel free to try NetworkManager – you can always switch back to dhcpcd if you have problems.

To switch to NetworkManager, just open a Terminal window and type:

```
sudo raspi-config
```

This launches the configuration tool. Go into option 6, Advanced Options, and then into option AA, Network Config – choose option 2, NetworkManager, and then reboot when prompted.

Once rebooted, NetworkManager should be running – to check, open the wireless LAN menu on the taskbar. It should now include an additional sub-menu at the bottom, Advanced Options – this is where you can connect to a hidden network, create a hotspot, or set up VPN connections. There are also a couple of useful information dialogs which allow you to see and change parameters for any network connection already configured.

Once you have switched to NetworkManager, you will need to reconnect to any wireless network to which you were connected under dhcpcd – connections do not automatically transfer between the two systems. Similarly, if you have made any customisations to a wired connection, such as setting a static IP address, you will need to redo those customisations under NetworkManager. (Most of these settings can be accessed via the connection editor, accessed from the 'Edit Connections' option in the Advanced Options sub-menu.)

In order to use a VPN, you need to install the relevant VPN plug-in. The OpenVPN plug-in is useful for many networks. To add it, open a Terminal window and type:

```
sudo apt install network-manager-openvpn-gnome
```

When this has installed, choosing the 'Add VPN Connection' option from the Advanced Options menu will open a dialog offering OpenVPN as a connection type.

If you find that NetworkManager is causing problems, you can use raspi-config to go back to dhcpcd – just enter the Network Config option as described above, and choose option 1, dhcpcd.

How to upgrade

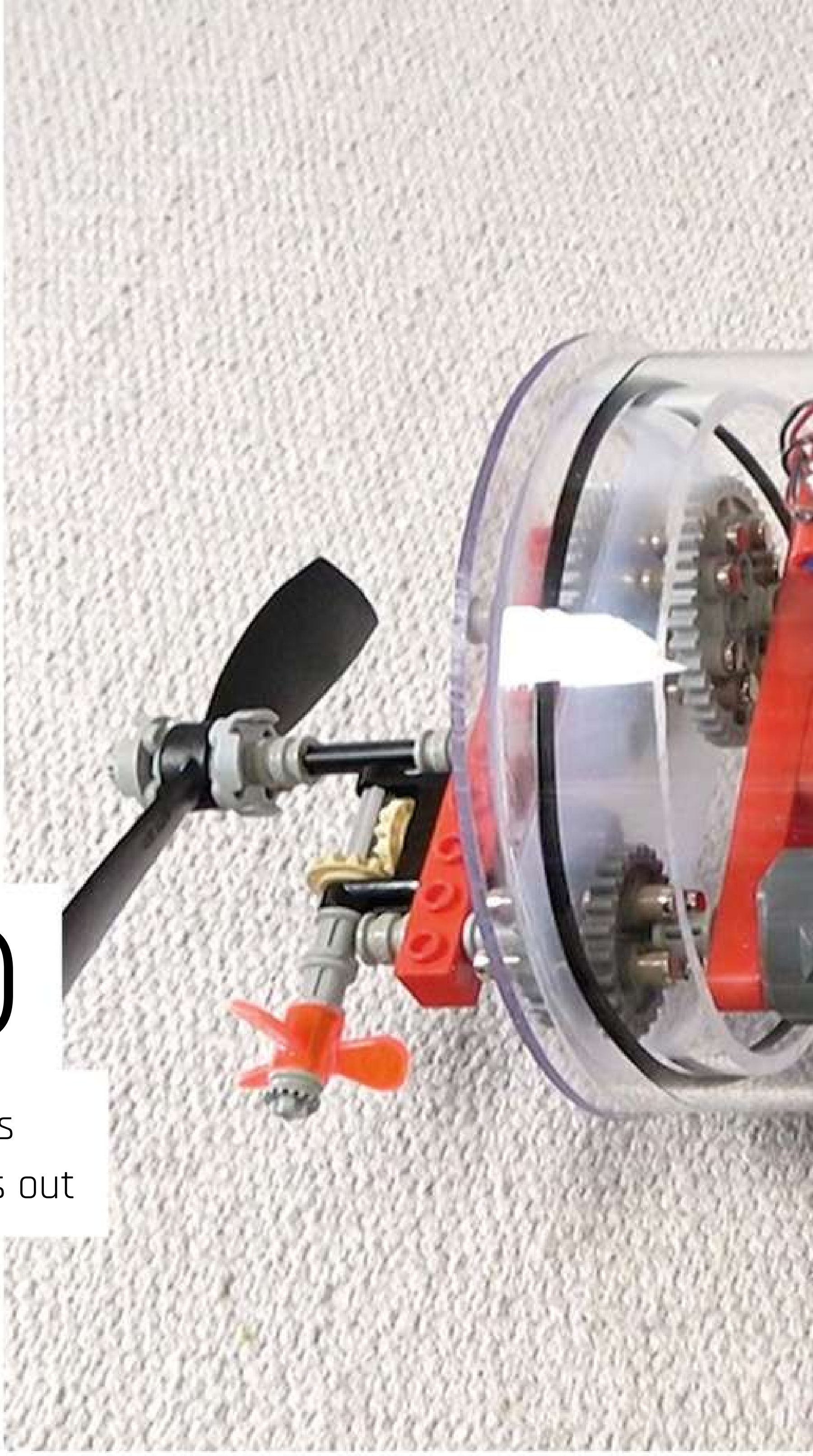
The new image is available for download from Raspberry Pi's downloads page (**magpi.cc/downloads**). It can also be flashed straight to an SD card using Raspberry Pi Imager (**magpi.cc/imager**). To update an existing image, use the usual Terminal command:

```
sudo apt update
sudo apt full-upgrade
```

Then, in order to load the new taskbar plug-ins, launch Appearance Settings from the Preferences section of the main menu, and on the Defaults tab, press the button corresponding to the size you prefer. Then reboot – this will reload the taskbar and load the new plug-ins.

To install NetworkManager on an existing image, open a Terminal and type:

```
sudo apt install network-manager
```



LEGO Submarine 4.0

The challenge of keeping a LEGO-based submarine afloat is solved with Raspberry Pi Zero W, as **Rosie Hattersley** finds out

MAKER

Brick Experiment Channel

Software engineer BEC has a lifelong passion for designing builds using LEGO Technics parts. Raspberry Pi has proved invaluable to their most recent projects.

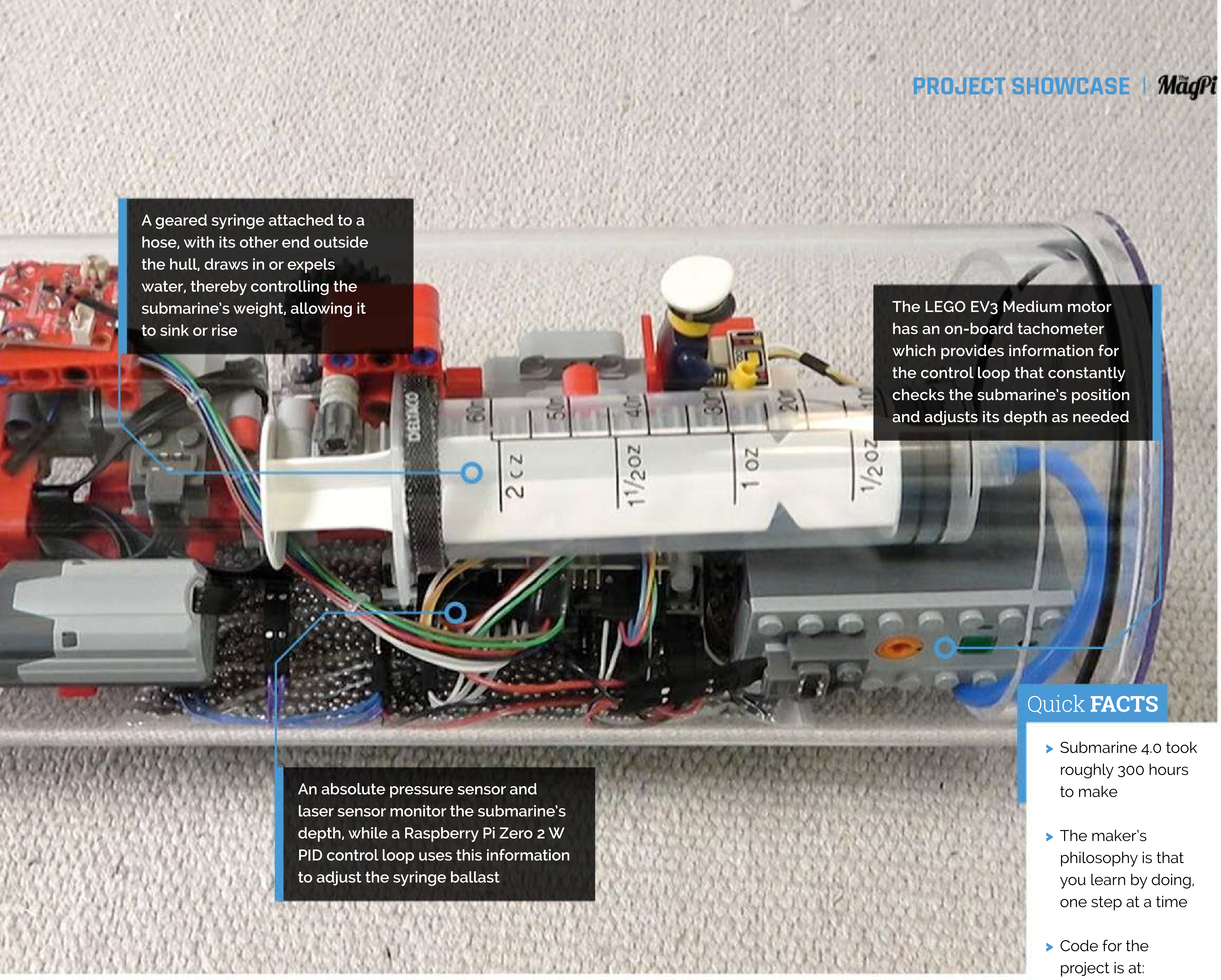
magpi.cc/brickexperiment

This LEGO builder extraordinaire is a "middle-aged guy from Finland" who dreams up, and then painstakingly creates, working models that address complex physics and mathematical challenges. His latest build, LEGO Submarine 4.0, runs off a LEGO EV3 motor and Raspberry Pi Zero 2 W. Like all his builds, the submarine project is thoroughly documented on his YouTube channel, Brick Experiment Channel (**magpi.cc/becyoutube**), where his methodical approach to construction and problem-solving have earned him more than 2.7 million subscribers.

This is the fourth LEGO submarine design he has completed. The first three variously used propellers to add or reduce buoyancy ("gravity and buoyancy stay always the same while the propellers exert force"); a balloon, and an air compressor to adjust the amount of water displaced, thereby controlling whether the submarine sinks or rises; and a piston ballast to suck in more water to add weight and increase the sub's gravity.

He settled on the last method for Submarine 4.0, despite the difficulty of gauging the neutral buoyancy point. However, it had proved a more stable setup and would not compress under pressure when submerged. Most importantly, "you

A submarine needs a captain, which was easily supplied in the form of LEGO part ID col154



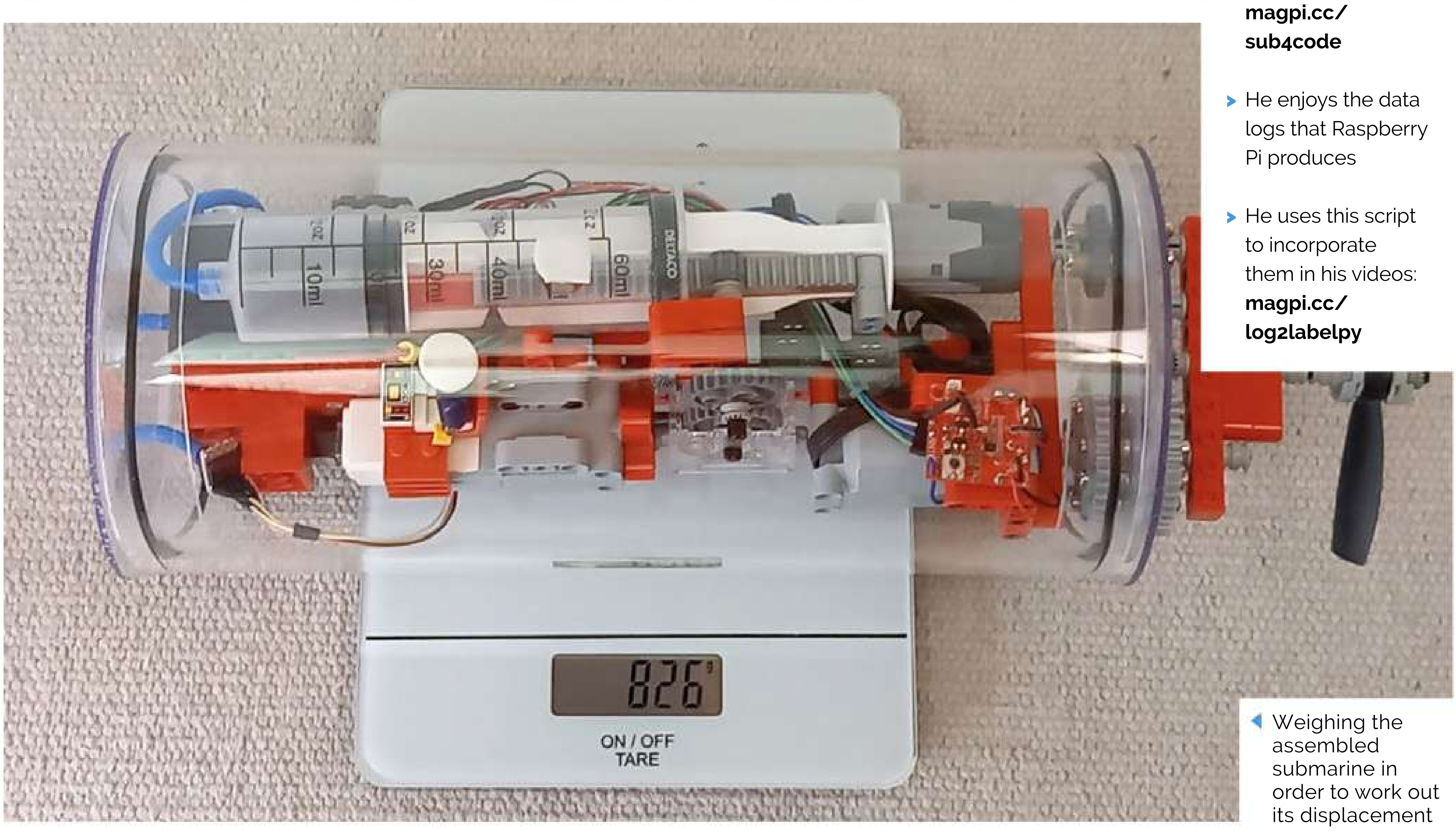
A geared syringe attached to a hose, with its other end outside the hull, draws in or expels water, thereby controlling the submarine's weight, allowing it to sink or rise

The LEGO EV3 Medium motor has an on-board tachometer which provides information for the control loop that constantly checks the submarine's position and adjusts its depth as needed

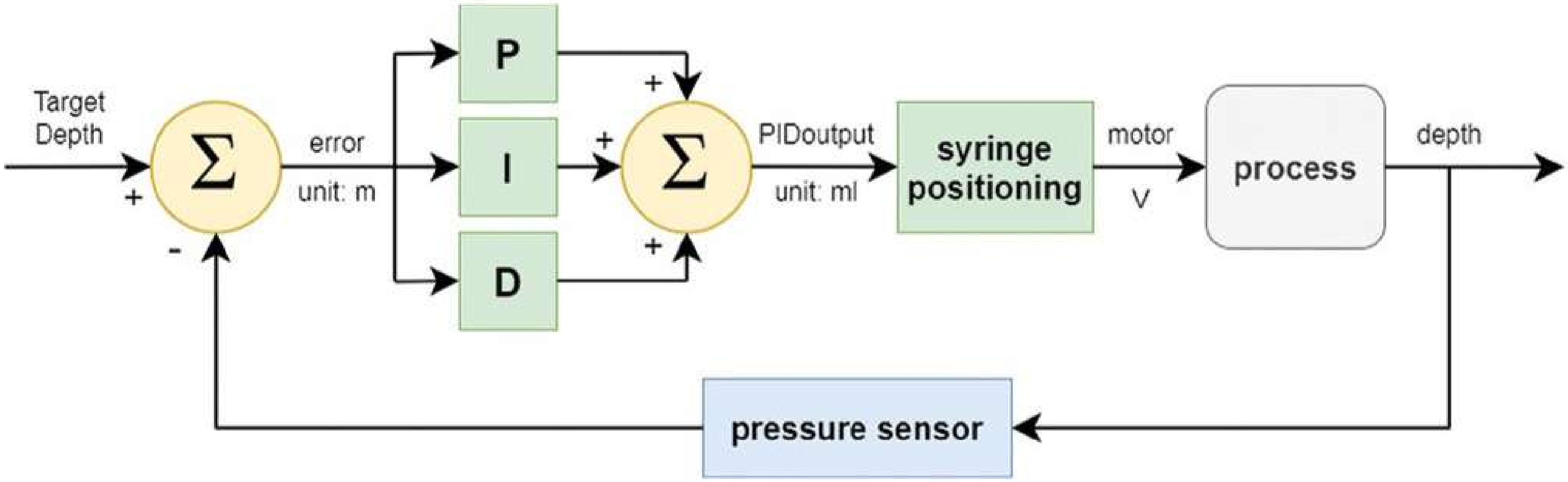
An absolute pressure sensor and laser sensor monitor the submarine's depth, while a Raspberry Pi Zero 2 W PID control loop uses this information to adjust the syringe ballast

Quick FACTS

- Submarine 4.0 took roughly 300 hours to make
- The maker's philosophy is that you learn by doing, one step at a time
- Code for the project is at: **magpi.cc/sub4code**
- He enjoys the data logs that Raspberry Pi produces
- He uses this script to incorporate them in his videos: **magpi.cc/log2labelpy**



Weighing the assembled submarine in order to work out its displacement



▲ Raspberry Pi's PID control loop takes depth information from the sensors and uses it to adjust the syringe ballast

> He attached a magnet on the inside top of the submarine frame so he could 'fish' the craft out of danger

can measure the piston position with a LEGO EV3 motor that contains a tachometer. That will help the control loop." This loop is also the reason for using Raspberry Pi: as we reported last issue, he had recently made an impressive inverted pendulum that uses a PID (proportional integral derivative) control loop running on Raspberry Pi Zero 2 W to accurately measure and compensate for constantly changing speed, location, and pressure levels.

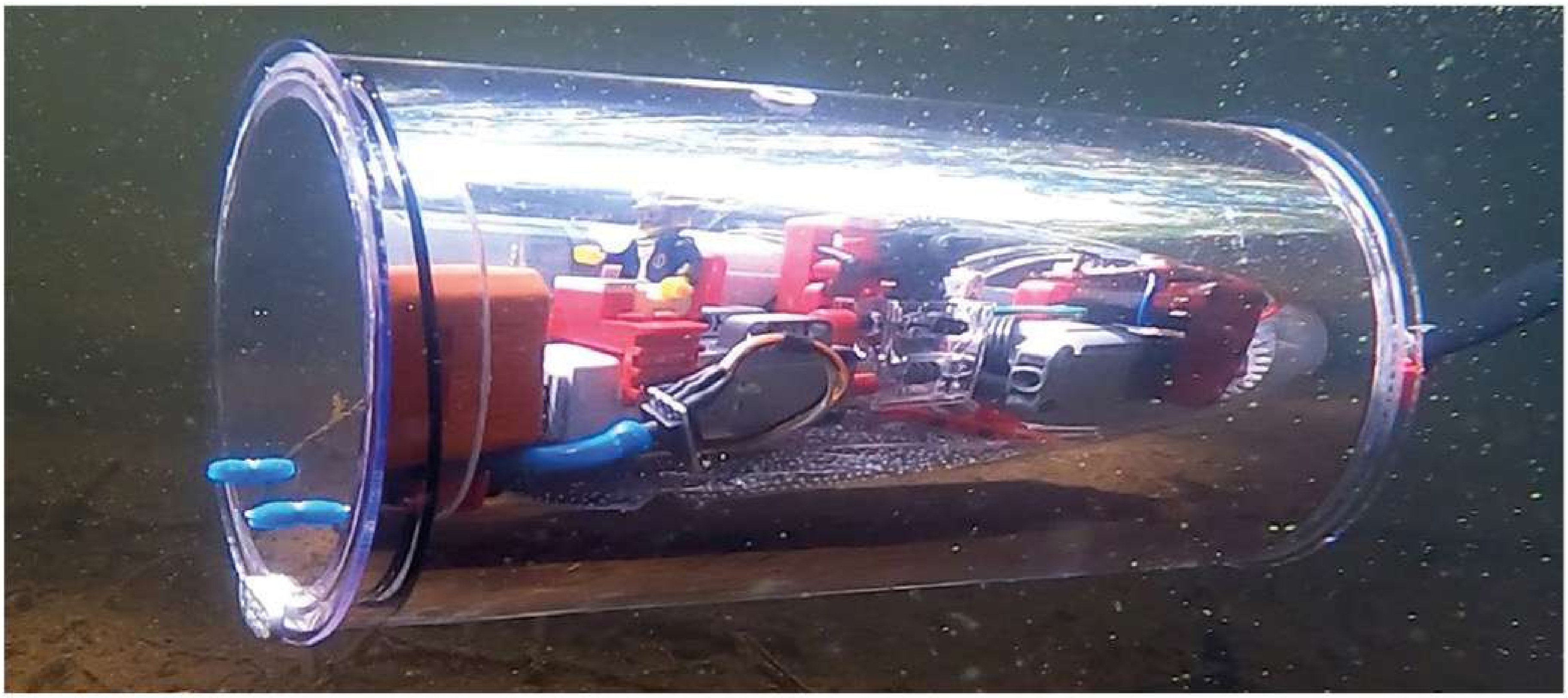
He planned a similar setup here, to monitor and control the submarine's depth. He soon found the wireless LAN connectivity invaluable when tweaking the PID parameters and updating any Python code without having to physically connect to the Pi Zero 2 W via USB (which would have involved carefully extracting everything from the precision environment he'd created). He now says wireless LAN is "an absolute necessity" which "made the development process a lot faster."

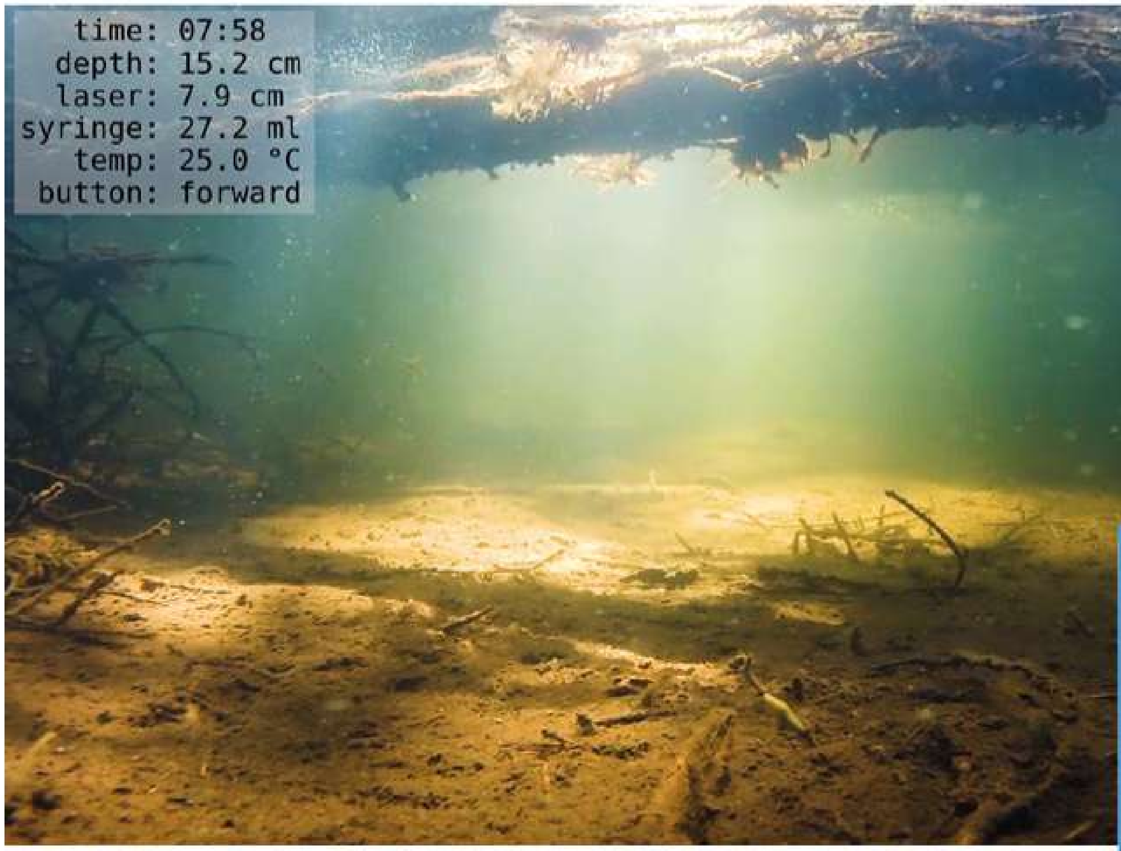
Underwater flying machine

A fair amount of time and effort went into creating the submarine's beautiful transparent acrylic case with tightly fitting and invisible end caps. The basic acrylic cylinders were precision-cut and end pieces with rubber seals attached to form a waterproof unit. LEGO gears were fitted to control the syringe that would adjust the buoyancy.

An absolute pressure sensor (which measures pressure relative to a vacuum, and is unaffected by the ambient pressure) is used to track the submarine's depth. It connects to Raspberry Pi Zero 2 W via I2C. A SparkFun TFMini-S Micro laser sensor provides a second means of measuring the submarine's depth, but its accuracy is affected by the murky environments in which he was using it.

▼ A rigorous series of underwater tests saw Submarine 4.0 submerged in a murky river





River testing data with log info provided by Raspberry Pi

A more successful purchase was the radio board he harvested from a cheap Chinese toy submarine, having chosen it for its 27MHz radio frequency (needed to penetrate water) and its aesthetically pleasing controller. Raspberry Pi provides enough juice to power the board, so he decided to discard its LiPo battery in favour of a LEGO waterproof rechargeable battery pack.

Diving for pearls

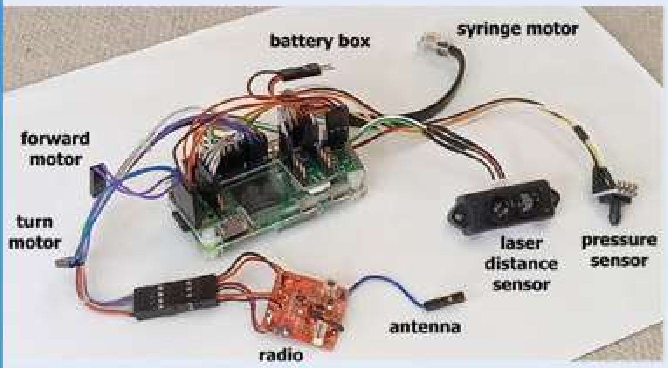
In Submarines 2.0 and 3.0, he used lead pellets to provide extra weight, but they were quite sizable and took up valuable space inside the submarine's frame. For this version, he splashed out on expensive 2.5 mm tungsten pellets weighing 18 g/cm^3. Weighed on a kitchen scale, the submarine was 826 g, with a displacement of 1614 g. He added 580 g of tungsten pellets to make the submarine dive gently, making adjustments using the syringe.

To prevent entanglements while on manoeuvres, he attached a magnet on the inside top of the submarine frame so he could 'fish' the craft out of danger, if needs be. With lots of weeds and obstructions on the river bed, he was keen to avoid collisions, especially as it cost more than 600 EUR.

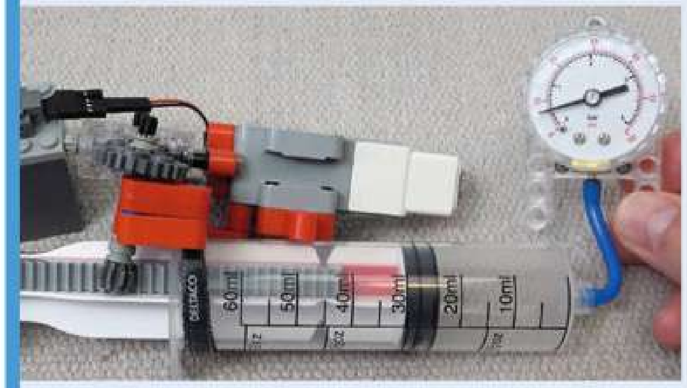
Thankfully, Submarine 4.0 has performed well in a range of environments from swimming pools and water tanks to a nearby river. "It drives well under water. The automatic depth control really makes controlling it easy, as you can focus on pressing only forward/backward and left/right buttons and forget the dive/surface buttons. I'd say the controls are as good as in Submarine 2.0, which has been the best so far," he reports.

Nonetheless, as a perfectionist, he notes several areas for improvement.

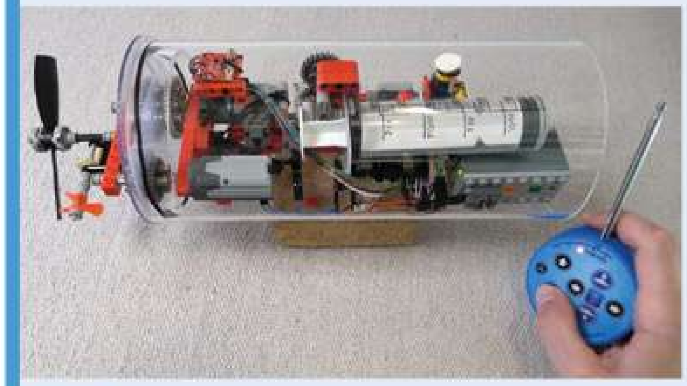
Underwater explorer



01 As well as a food syringe and LEGO gear wheels, you'll need a Raspberry Pi Zero 2 W with 16GB microSD card, a LEGO EV3 Medium motor with tachometer, and a 27MHz radio controller.



02 A motorised LEGO gearing system adjusts the ballast syringe. A hose sucks in water and draws water into the syringe chamber. Buoyancy remains constant while the extra weight of water increases the gravitational pressure.



03 A radio board and controller culled from a cheap toy provide a user-friendly remote control, powered by Raspberry Pi, with no need for a dedicated battery.

Pico Pocket Gamer

A pocket console that's simple enough for a beginner to build – now that's **Nicola King's** kind of project

MAKER

Grgo Mariani

Grgo is a software developer working for IBM in Böblingen, Germany, and a huge Raspberry Pi fan.

magpi.cc/picopocket

While reminiscing about a retro electronic game called Lights Out (a puzzler with a grid of lights), Grgo Mariani realised he could make his own updated version with an unused Raspberry Pi Pico.

"I remember how I used to play it while visiting my family and think about working out a perfect strategy to solve each new puzzle. Since I recently became a father, I wanted my son to have this experience as well, but unfortunately, I could not find a shop where I could buy this product anywhere," he recounts. This was the push he needed to start creating, and the result is a rather fun little games machine that is simple enough for a newbie.

Newcomers welcome

As the hardware encompasses just a Pico, a 320×240 touchscreen display, and five mechanical switches, Grgo sees this as a very achievable make. All that's required is connecting several Pico pins to the display and mechanical switches. "No resistors, no capacitors, no battery, no external power supplies… I would call this beginner-level," he confirms. While he designed a custom PCB to streamline his mini console, he says this is optional.

On the software side, he's created a simple engine to handle the key and touch inputs, update the game state, and draw to the frame buffer in a loop. "Each game shares one 320×240 frame buffer for drawing and one 64×64 texture buffer. There is also a 16kB space used for game state memory," he explains.

So far, he has created 14 games for the device, including his versions of Snake, Pong, and of course, Lights Out, and took a flexible approach to game development. "The end design became much more than that one single game, as I really took an agile approach while building it."

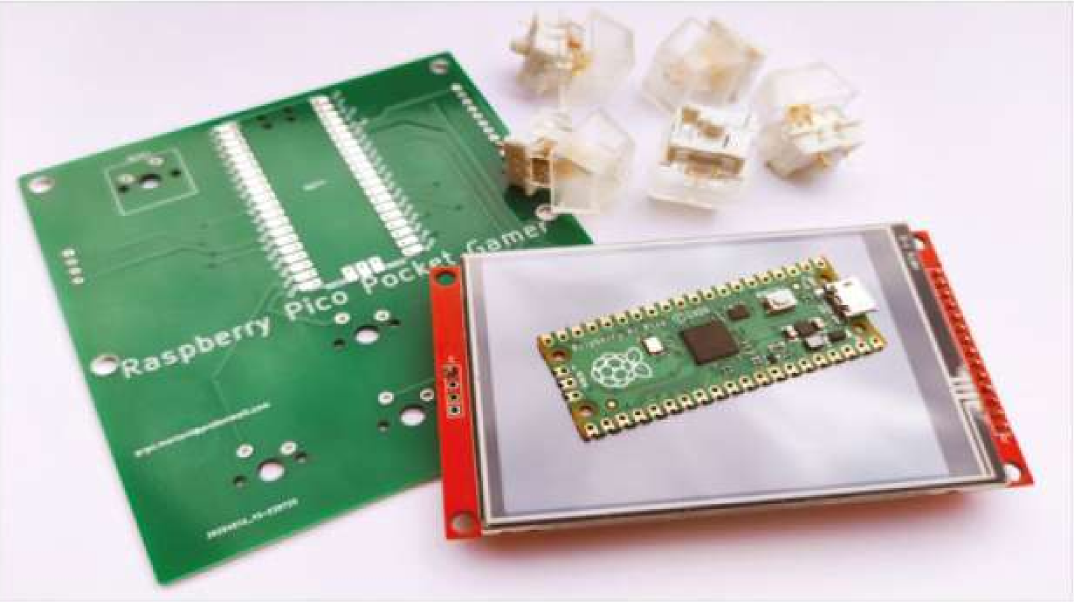
He wrote most of the open-source code himself while modifying a few parts from other projects – in particular for handling communication with the display. "I also adapted the ray casting algorithm [for] my Labyrinth game." He emphasises that, from a coding perspective, he's not using any advanced capabilities of Pico here.

Apart from finding the time, as a new parent, to spend on his project, Grgo found that the most difficult element was choosing the right display and constantly rewiring the breadboard to adapt to the new pinouts. "It took me several months to find the right display… PCB design is also something I never did before, so I had to learn it from scratch."

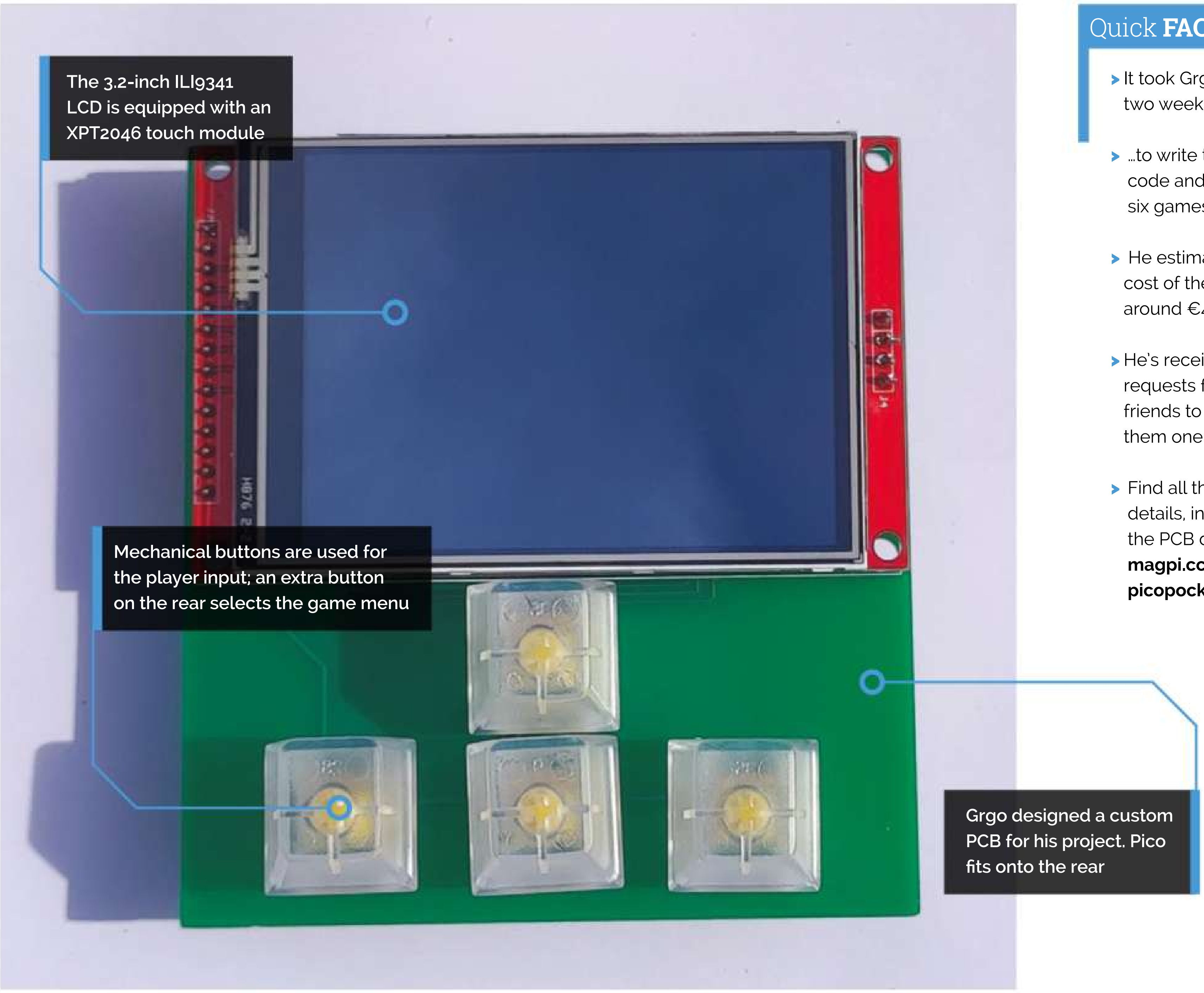
Take a look at his useful YouTube video (**magpi.cc/picopocketyt**), which shows him putting the device together.

Favourable feedback

The response received around the Pico Pocket Gamer has been super-positive, and Grgo's been bowled over by the reaction by some members of the Reddit community. "Some people already DM-ed to ask me how they can replicate the project. It brings me joy to think that I might have



▼ All the parts for the project, pre-assembly; it's fairly simple to put together



The 3.2-inch ILI9341 LCD is equipped with an XPT2046 touch module

Mechanical buttons are used for the player input; an extra button on the rear selects the game menu

Grgo designed a custom PCB for his project. Pico fits onto the rear

Quick FACTS

- It took Grgo around two weeks…
- …to write the main code and first six games
- He estimates the cost of the build at around €40
- He's received requests from friends to build them one!
- Find all the details, including the PCB design, at: **magpi.cc/picopocket**

ignited a spark in someone to give their Pico, and some new project they have in mind, a try."

As for future projects, Grgo doesn't have any concrete plans yet, but confirms, "it will have something to do with the Pico W, that's for sure." One thing he would really like to do is to implement some of the algorithms from Michael Abrash's *Graphics Programming Black Book*, and make a simple 3D game using them. He plans on adding more games to this console, and has been buoyed along by the momentum of creating each game: "With every game finished, there was this drive to build just one more." And surely that's a sentiment that many makers can empathise with.

> "The end design became much more than that one single game"

Running a game on the Pico Pocket Gamer – it can even handle 3D graphics

Chasing Your Tail

Turning the tables on trackers, a Raspberry Pi 3-based device that lets you know if you're being followed, hears **Rosie Hattersley**

MAKER

Matt Edmondson

Digital forensics specialist for the US Department of Homeland Security and keen maker and tinkerer.

digitalforensicstips.com

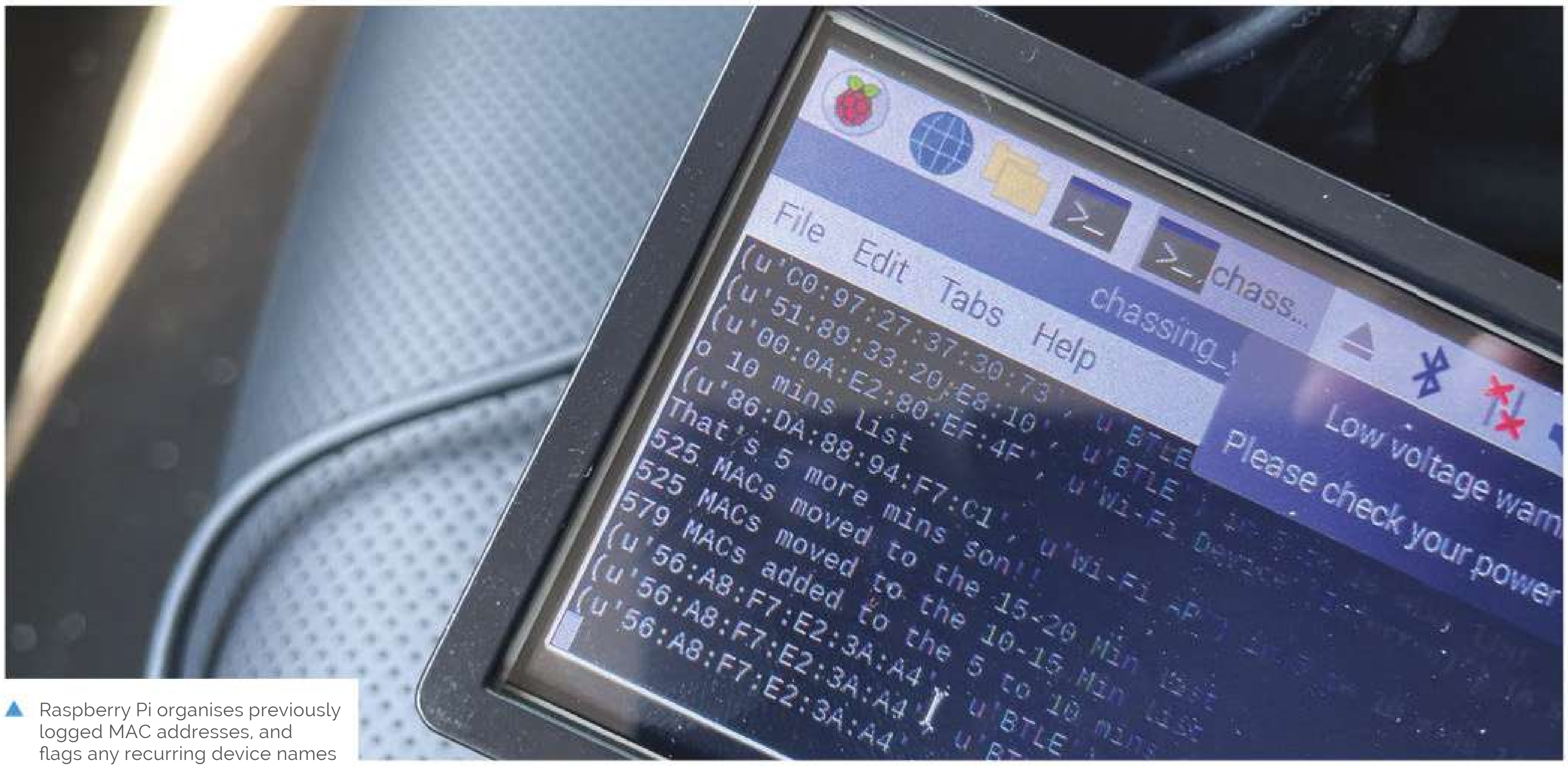
Even spooks get spooked if their Spidey sense tells them they're being followed. A conversation with an intelligence handler concerned for their client's safety led US government digital forensics expert Matt Edmondson to put together a Raspberry Pi 3-based alert system to warn of a potential tail.

In true espionage style, the Chasing Your Tail device fits inside an anonymous black briefcase and silently flags an issue, should a tracker be identified. It looks for smartphones and other Bluetooth or WiFi-enabled devices that are actively scanning for connections – essentially any modern phone, laptop, or tablet – and lets the anxious 'mark' know that the same device has been in close proximity for the past five, ten, or twenty minutes. Assuming the device that's flagged isn't on your 'ignore' white list because it belongs to a travelling companion or colleague, there's a chance it might belong to someone following you.

Chasing Your Tail creator Matt notes that there are plenty of options if you want to track someone's movements or digital activities, but few ways to identify whether it's happening and pinpoint who's behind it. "It's very depressing to see how many devices are out there to spy on people, versus how many devices detecting that people are being spied on."

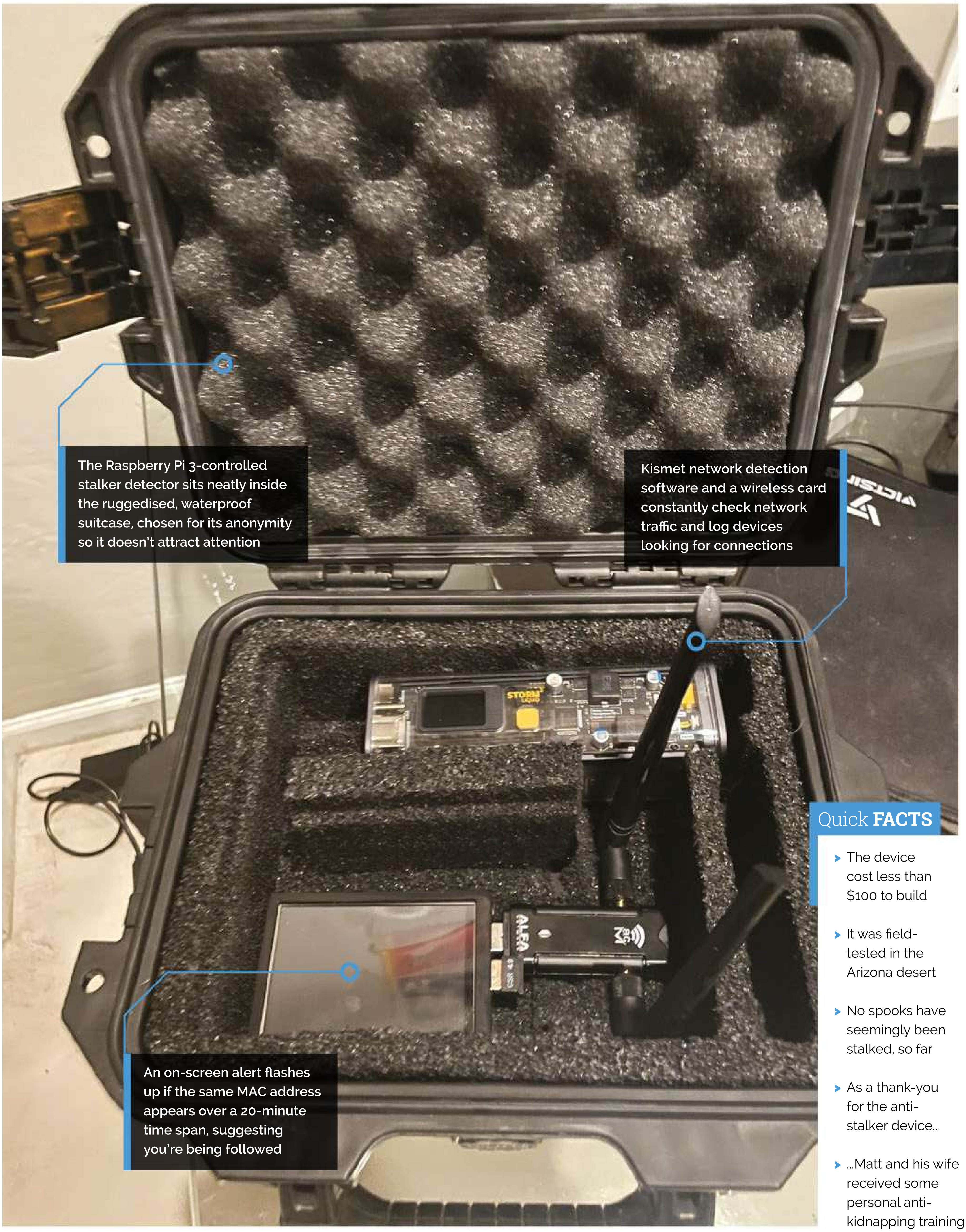
Makers gonna make

Matt works for the US government with a focus on open-source intelligence, and recently gave a talk on his anti-stalking device at the Black Hat security conference: **magpi.cc/chasingyourtail**.



```
(u'C0:97:27:37:30:73', u'BTLE
(u'51:89:33:20:E8:10', u'BTLE
(u'00:0A:E2:80:EF:4F', u'Wi-Fi Device
o 10 mins list
(u'86:DA:88:94:F7:C1', u'Wi-Fi AP
That's 5 more mins son!!
525 MACs moved to the 15-20 Min list
525 MACs moved to the 10-15 Min list
579 MACs added to the 5 to 10 mins
(u'56:A8:F7:E2:3A:A4', u'BTLE
(u'56:A8:F7:E2:3A:A4'
(u'56:A8:F7:E2:3A:A4', u'B
```

▲ Raspberry Pi organises previously logged MAC addresses, and flags any recurring device names



The Raspberry Pi 3-controlled stalker detector sits neatly inside the ruggedised, waterproof suitcase, chosen for its anonymity so it doesn't attract attention

Kismet network detection software and a wireless card constantly check network traffic and log devices looking for connections

An on-screen alert flashes up if the same MAC address appears over a 20-minute time span, suggesting you're being followed

Quick FACTS

- The device cost less than $100 to build
- It was field-tested in the Arizona desert
- No spooks have seemingly been stalked, so far
- As a thank-you for the anti-stalker device…
- …Matt and his wife received some personal anti-kidnapping training!

He's not your typical security expert, though. Matt first became interested in technology aged 10 when, a year before she died, his mum bought him a TRS-80 computer as a Christmas present and he began to learn BASIC.

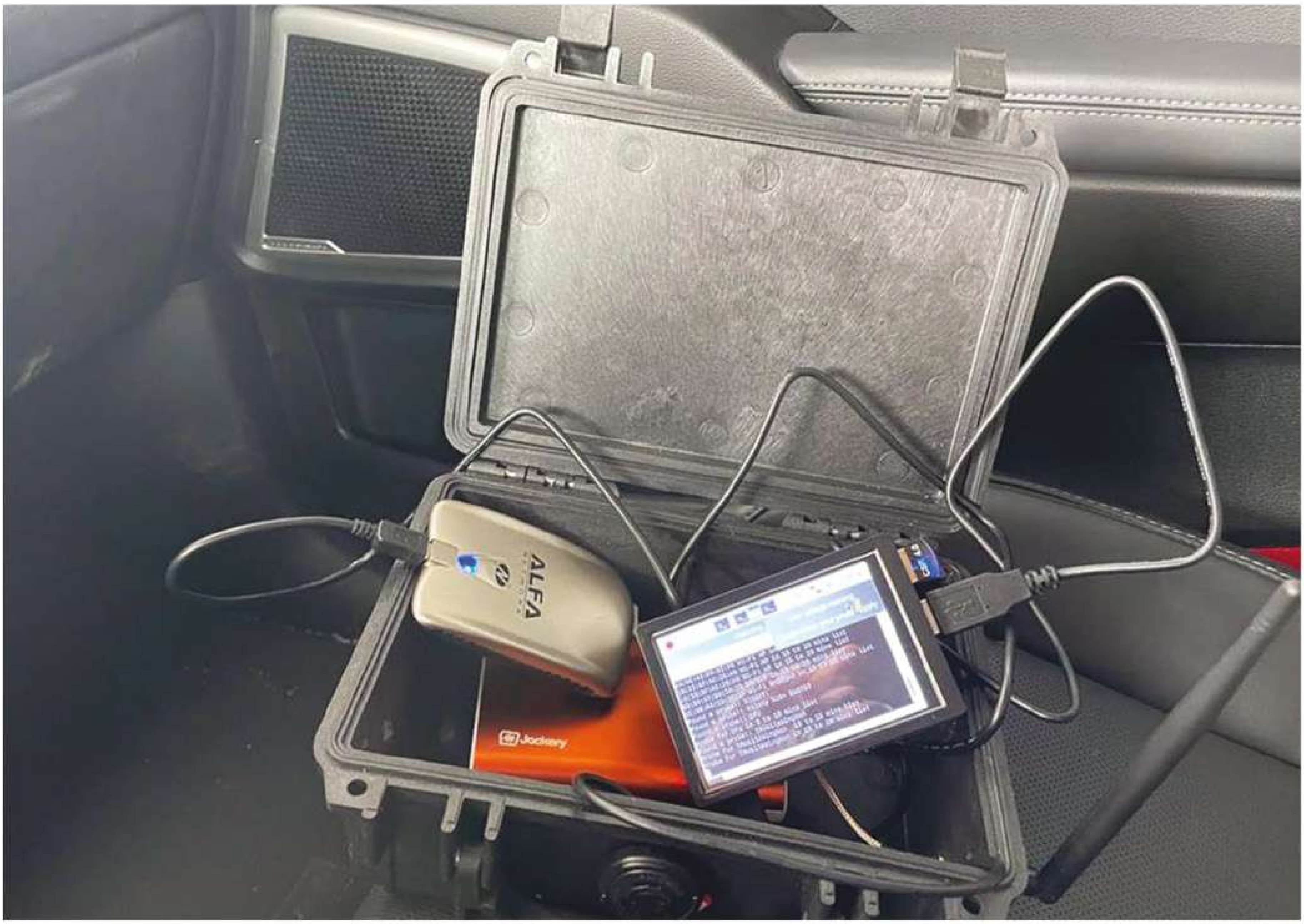
Computing remained a hobby when he began a career in law enforcement, but his self-taught skills came to the fore as he began reading up on how to fix and connect various office devices and equipment, "building things that needed to be built," eventually leading to Matt being recruited as a technical agent in a government department. He credits not having a formal computing education as one of the reasons he has no fear of computers, since he's never been aware of what boundaries there are supposed to be. Nonetheless, Matt eventually learned MySQL databases, PHP, and GIS mapping, developing a strong interest in digital forensics. A decade ago, this led to him developing and teaching an OSINT (open-source intelligence) class, including a sandbox activity simulating possible security breach risk scenarios. At this point "a light switch flipped on about the need for far more situational awareness," he explains.

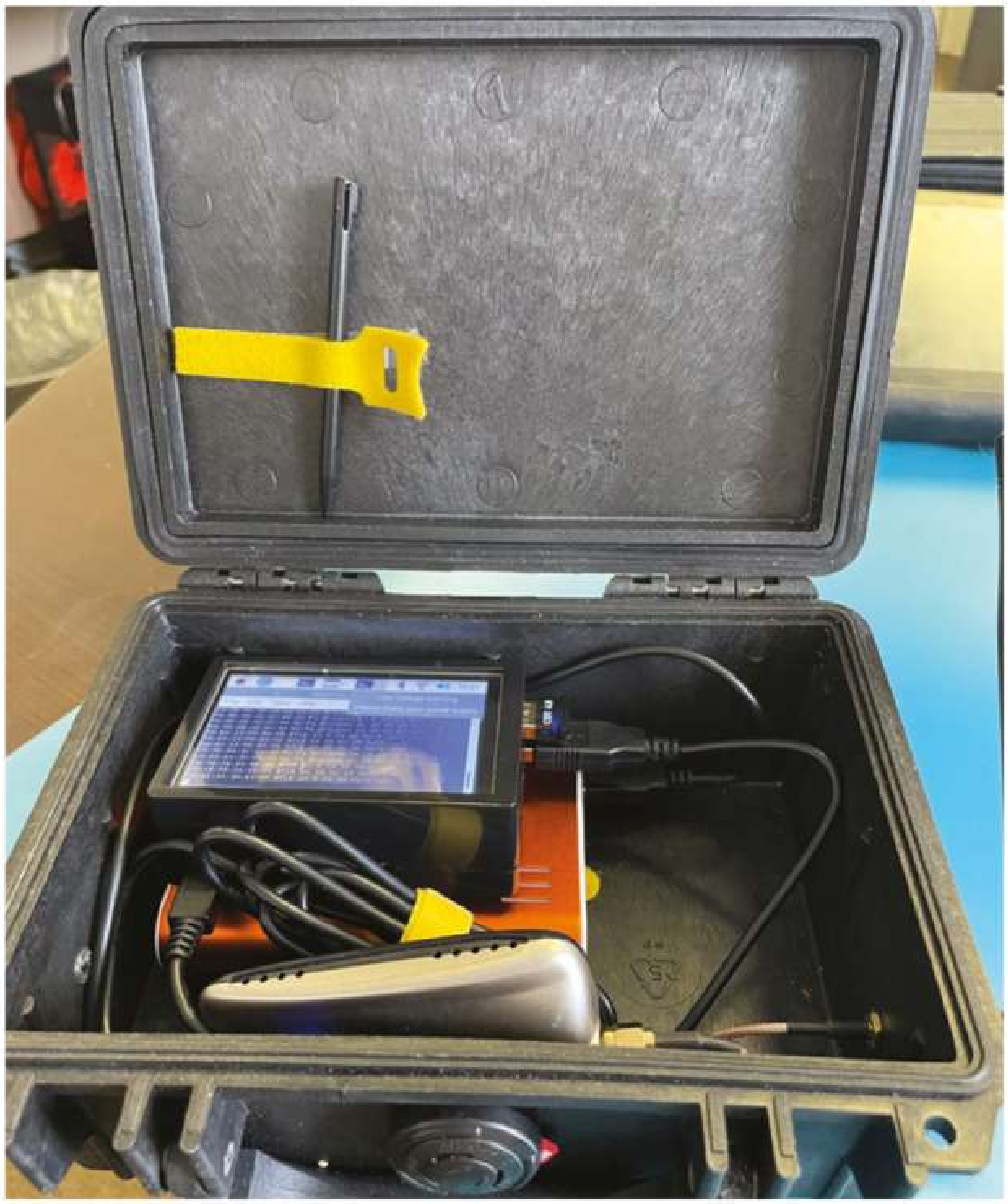
A second set of eyes

Having spent 21 years working in digital forensics for the US government, Matt is keenly aware of the dangers that covert operations bring to both handlers and those being protected. Even the most security-conscious person carries a smartphone, however, and these can easily betray someone's location by dint of the way they're designed to constantly look for available connections.

The challenge is not trying to give away your own location, he warns. "If I was trying to work out if someone was following me, I would go to three different places: I'd go grab a coffee, then maybe put gas in my car, then maybe go to a bookstore, and then I would look at and see, did I see any devices at all three locations?"

▼ The deliberately pared back viewer provides at-a-glance alerts if someone is potentially tailing you as you drive





Matt field-tested his device in the Arizona desert, but admits urban environments present a bigger, more complex challenge

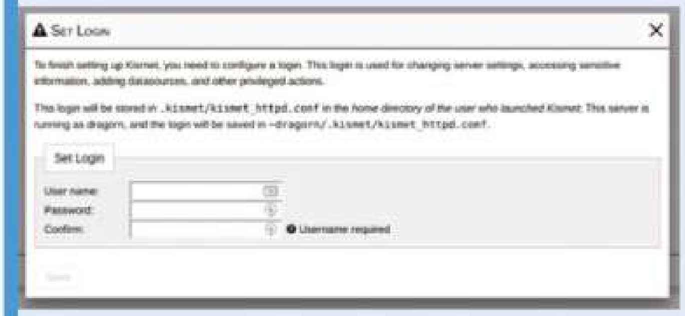
Years later, an attendee at that OSINT class from a different agency contacted him with concerns. "He wasn't worried about his physical safety; he was worried about the physical safety of this person he was talking to, and meeting with," and wanted a second set of eyes. The man was after a physical detection device to use alongside the tradecraft and surveillance detection tools he already had.

> In true espionage style, the device fits inside an anonymous black briefcase and silently flags an issue, should a tracker be identified

Given his history of making things, Matt already had the hardware he needed, including a pair of Raspberry Pi 3 computers, barring a cheap $25 display he bought online. "I can write ugly Python code that gets the job done usually." A month later, Matt handed over the suitcase containing the promised anti-stalker device.

Sniff out spies

Being stalked can cause life-affecting anxiety and stress. For this reason, Matt's made the full code for his anti-stalking device available on GitHub: **magpi.cc/cytgit.**



Set Login

To finish setting up Kismet, you need to configure a login. This login is used for changing server settings, accessing sensitive information, adding datasources, and other privileged actions.

This login will be stored in .kismet/kismet_httpd.conf in the home directory of the user who launched Kismet. This server is running as dragorn, and the login will be saved in ~dragorn/.kismet/kismet_httpd.conf.

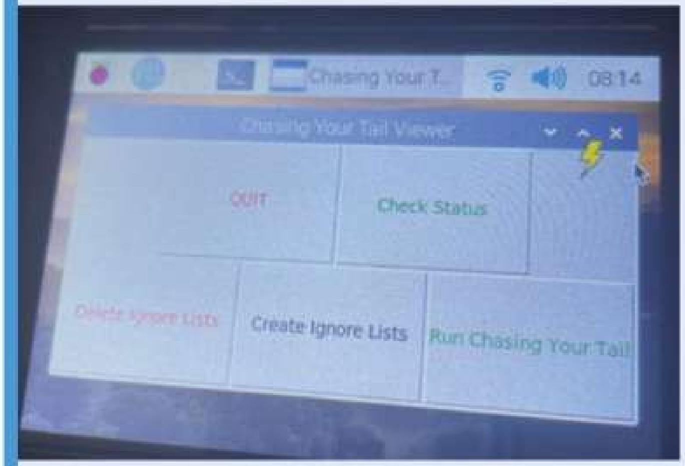
Set Login

User name:

Password:

Confirm: ● Username required

01 On a Raspberry Pi 3 or 4, update Apt and install the Kismet network analysis software (**kismetwireless.net**). Open a Terminal window in Raspberry Pi OS and install Kismet (**kismetwireless.net**).



02 Follow the instructions at **magpi.cc/cytprereqs** to install all the required packages. Set up your Raspberry Pi as a known Kismet device by typing and check the pi user has been added to the groups. Create a logging directory to store your logs. Then to make the display and Raspberry Pi autostart.

The New (i.e. Really Old) SDR: Surveillance Detection Route

03 Set up and enable the wireless card, and enable monitor mode with crontab. Reboot and use the Create Ignore Lists option to add any known networks or devices to the whitelist so they aren't flagged as possible devices tracking you.

Serial Typewriter

Riley Groeschel has taken an electronic typewriter from the late 1980s and turned it into a Linux terminal, as **David Crookes** explains

MAKER

Riley Groeschel

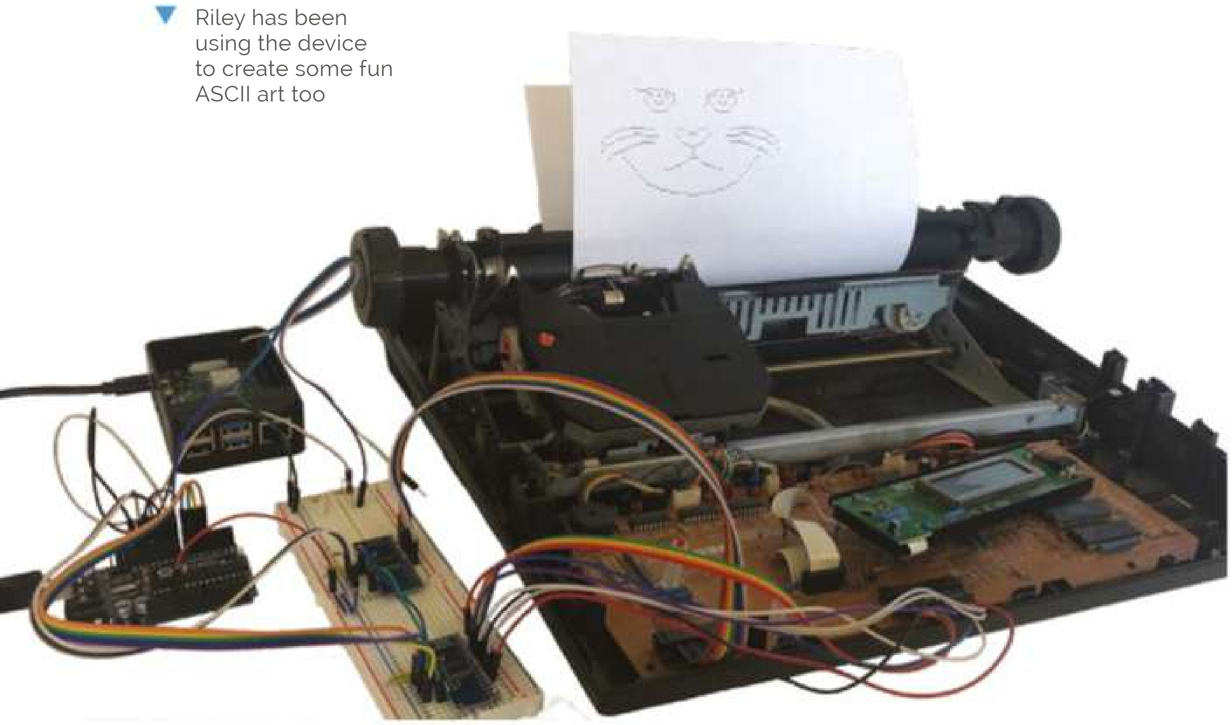
Riley Groeschel is a recent Mechanical Engineering graduate of Johns Hopkins University in Baltimore, Maryland. He loves breathing life into old electronics.

magpi.cc/serwriter

It may surprise you to know that some people still use typewriters today. There are government departments which tap away on them because they can't be hacked and they are used in some capacity by banks, prisons, businesses, and law firms.

Here at *The MagPi* magazine, we're also taken in by a typewriter's charm; from its immediacy and feel to the sound made as the letters are hammered on to the carriage. So, when Riley Groeschel decided to dust down a Brother AX-25 electronic typewriter and give it a new lease of life, we felt it was something to write home about.

"I've turned the typewriter into a Linux terminal," Riley says, revealing that it wasn't actually the original proposal. "Back in high school, my friend was working on a computer using a Z80 processor," he adds. "We planned on taking his finished computer and putting it inside a typewriter so we'd have something similar to a Commodore 64, but we gave up because he never got the computer working."



▼ Riley has been using the device to create some fun ASCII art too

Bashing out ideas

At that point, Riley considered using his typewriter shell to create a cyberdeck (a retro-looking futuristic computer). "That idea floated around my head until I saw CuriousMarc's video where he used a teletype as a Linux terminal and the idea was born," Riley says. "I knew I had to replicate the experience of using UNIX on a teletype like it was done half-a-century ago."

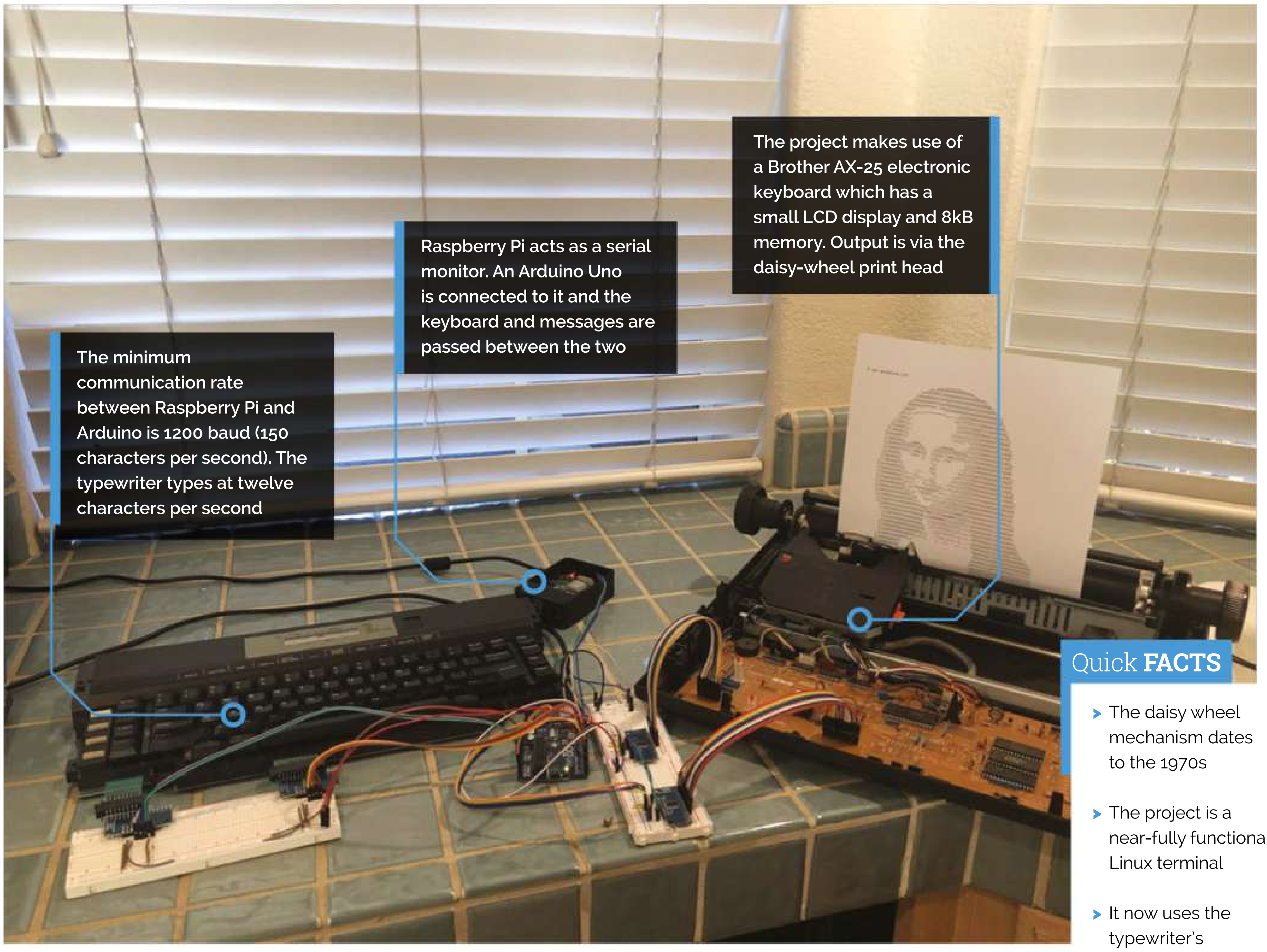
To do this, Riley decided to put a Raspberry Pi computer at the heart of the typewriter, coming up with a way to fool the old machine into believing a key had been pressed. The aim was to print the output from a Terminal window but, at this stage, a laptop was being used as opposed to the typewriter's own keyboard.

At first, Riley connected two multiplexers to the typewriter's keyboard 8×11 matrix and allowed them to be controlled by an Arduino Uno. "My Arduino is connected to my laptop over serial through a USB cable and I can send commands to it through the Arduino IDE's serial monitor," Riley explains.

"When the Arduino receives a message from my laptop, it forwards it to Raspberry Pi which has the serial code on its UART pins enabled. Raspberry Pi forwards it to the Arduino with the output of the command that it received. The Arduino then 'sends' it to the typewriter using the two multiplexers to emulate the typewriter's keyboard."

The right type

As commands or words are typed, they're banged straight out on to paper. Any resulting messages are also immediately printed. "I had to map out the typewriter's keyboard one key at a time by connecting each pair of pins on the two keyboard connectors together, and I put that all into an array which let me connect the ASCII characters from Raspberry Pi to the typewriter's 'keycodes'."



The minimum communication rate between Raspberry Pi and Arduino is 1200 baud (150 characters per second). The typewriter types at twelve characters per second

Raspberry Pi acts as a serial monitor. An Arduino Uno is connected to it and the keyboard and messages are passed between the two

The project makes use of a Brother AX-25 electronic keyboard which has a small LCD display and 8kB memory. Output is via the daisy-wheel print head

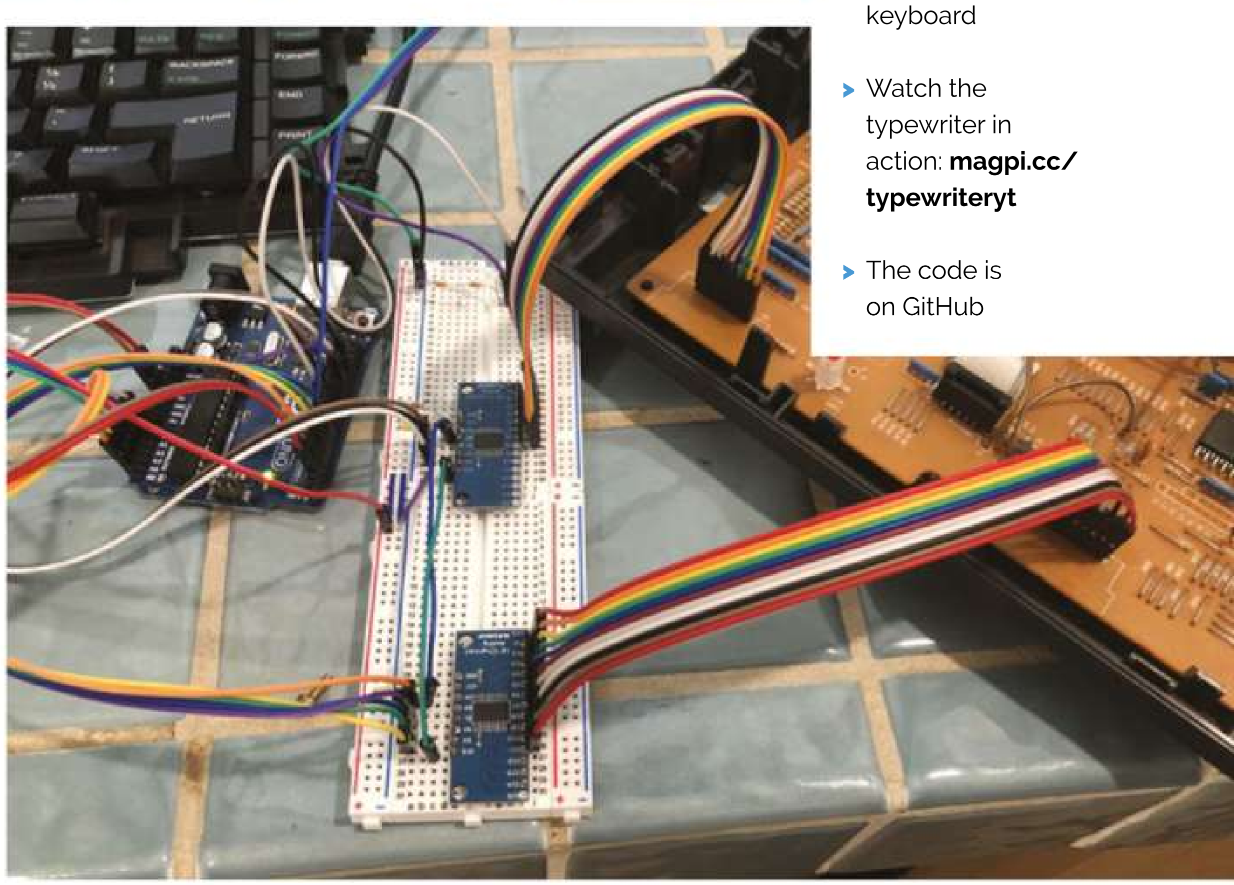
Quick FACTS

- The daisy wheel mechanism dates to the 1970s
- The project is a near-fully functional Linux terminal
- It now uses the typewriter's keyboard
- Watch the typewriter in action: **magpi.cc/typewriteryt**
- The code is on GitHub

After that was working, he connected the typewriter's own keyboard to the Arduino. "The Arduino checks each pair of pins one at a time using the multiplexers until it detects that a key has been pressed," Riley says. "Once it does, it sends that key to the Raspberry Pi computer for processing."

> " I knew I had to replicate the experience of using UNIX on a teletype "

Aside from working as a Linux terminal or as a teletype, the machine can also print large ASCII art, including a cool image of the Mona Lisa. Riley is also continuing to refine his project. "I plan on designing a custom circuit board that I'll install between the typewriter's keyboard and motherboard," he says. "I might even replace the typewriter's keyboard with a custom mechanical one to improve the typing experience."



▲ A voltage divider circuit has been created so that the 5V output of the Arduino is at a safe 3.3V level for Raspberry Pi

AuraLock

Forgot your keys? With Dillon McCardell and Erin Stanfill's project, you can simply walk right up to the door and watch it unlock, as **David Crookes** explains

MAKER

Dillon McCardell & Erin Stanfill

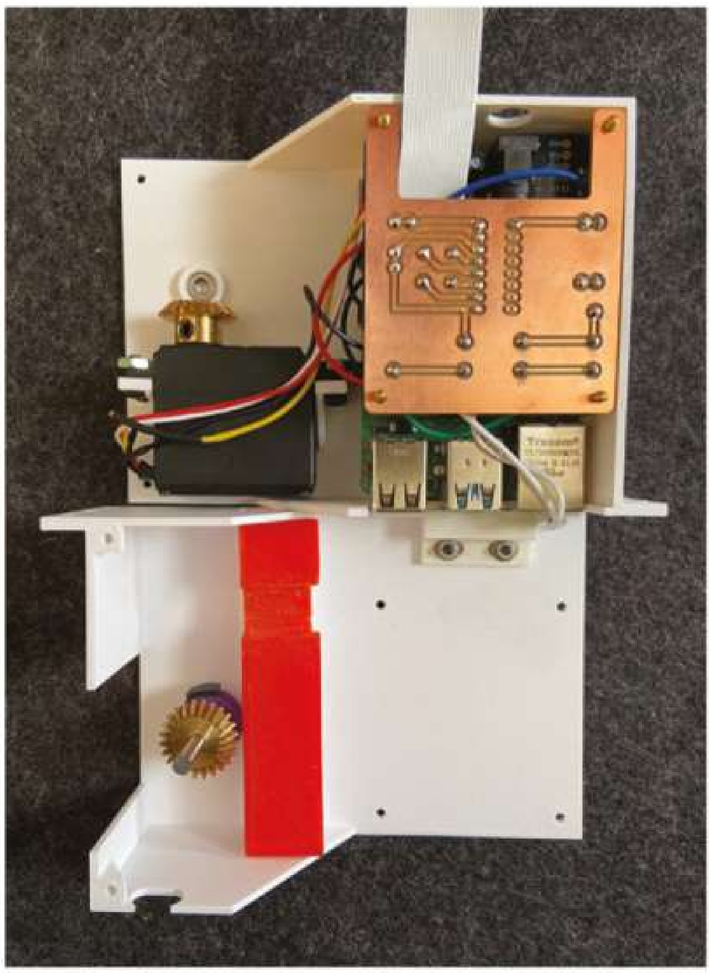
Dillon is a recent Computer Engineering graduate with a passion for design and development, taking ideas from concept to reality. Erin is a computer engineering graduate with a drive to continually learn, pushing technology into the future.

magpi.cc/auralock

Many of us use our face to unlock mobile phones and tablets, but imagine ditching your keys and doing the same to enter your home. It's that idea which underpins a project by Dillon McCardell and Erin Stanfill: a door locking system using facial recognition that uses off-the-shelf components.

Called AuraLock, it's primarily based around a Raspberry Pi 4 computer, a camera, door sensor, and Android mobile device. "Currently there are consumer facial recognition door locks, but they run on batteries and require push-button feedback to enable the facial recognition component," Dillon says. "This defeated the purpose of a hands-free unlocking feature."

Although enterprise-level facial recognition security systems can connect to automatic sliding doors, they are very expensive, so Dillon and Erin looked to create a go-between. "We wanted something to theoretically cost the same as current consumer facial recognition door locking devices," Dillon says. "It would be connected to your home's power and contain a motorised deadbolt, allowing the system to be powered 24/7 and completely hands-free." This, however, was not without its challenges.



Firebase's Python library was imported into software running on a Raspberry Pi 4 computer which, in turn, was fitted with a Servo PWM Pi HAT

Slide away

As well as needing to constantly search for a known face, the hardware module had to be mounted within the frame of the door. The deadbolt needed to be motorised so it could retract into the frame to unlock, and there needed to be a way of quickly and easily exiting a home without electronics getting in the way.

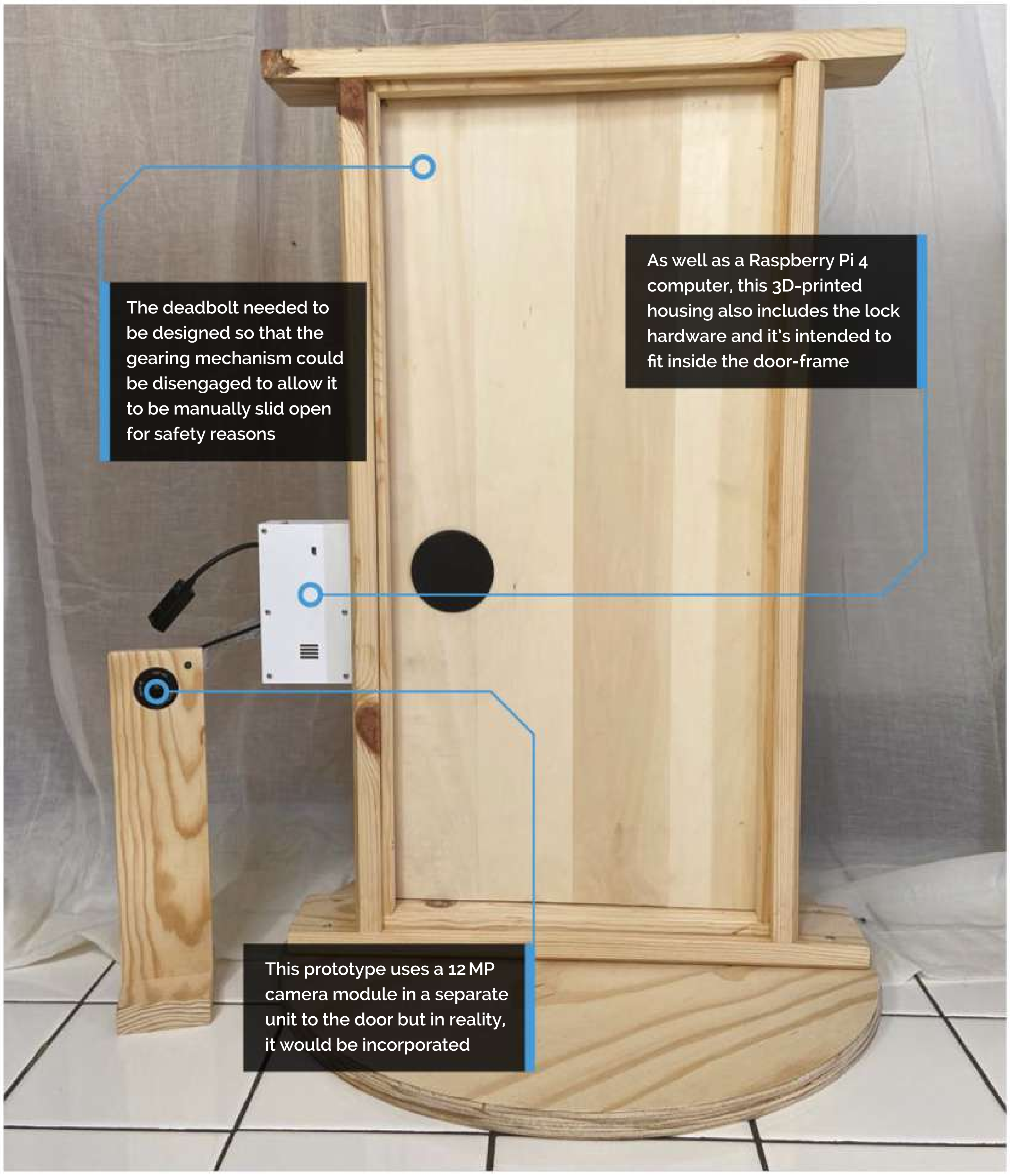
"I implemented a simple sliding mechanism to replace the knob on a standard door – it was able to remove the deadbolt with a single motion," Dillon says. "Other hardware features included the ability to unlock the door with facial recognition and an LED indicator on the exterior of the home to receive feedback. But other features, regarding usability and convenience, were primarily achieved in the Android mobile app."

These included remote unlock and the ability to add recognised faces. "We felt these two features would give the user a true smart home feel," Erin explains. "People would be allowed to enter the home even if the owner wasn't there, and adding faces gives the user the ability to adapt the door as needed. Yet, having never built an app before, the whole plan was a little intimidating."

Professional feel

The pair also faced problems when implementing OpenCV to recognise faces. "The library is vast and difficult to install easily," Dillon says. "Once OpenCV was installed, I realised it was not designed to accept new face encodings dynamically – all registered images were generally hard-coded on bootup."

To overcome this, Dillon implemented a dynamically updating dictionary, and he refreshed



The deadbolt needed to be designed so that the gearing mechanism could be disengaged to allow it to be manually slid open for safety reasons

As well as a Raspberry Pi 4 computer, this 3D-printed housing also includes the lock hardware and it's intended to fit inside the door-frame

This prototype uses a 12 MP camera module in a separate unit to the door but in reality, it would be incorporated

the OpenCV code to reflect the changes. The pair also used a Google Firebase database that allowed for communication between Raspberry Pi and the Android app. "Having to communicate with other devices through Firebase added a level of difficulty that wasn't anticipated at first," Erin says.

> "Having to communicate with other devices through Firebase added a level of difficulty"

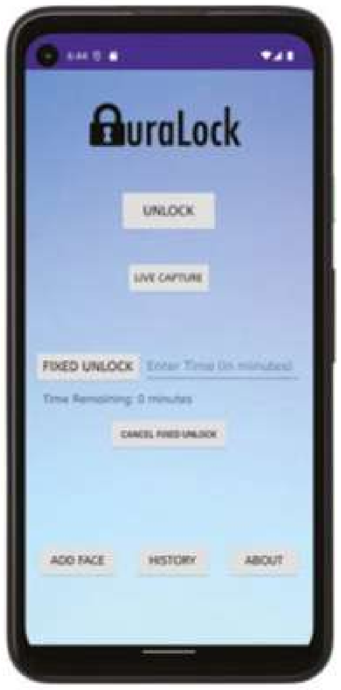
This was achieved by updating and reading database variables – they'd be updated when a new user unlocked the door, for example, and the mobile app would notice the change and show an image of the user who had activated the lock. "By implementing this method, we were able to transfer both variable values and images from Raspberry Pi to the mobile app and back again through Firebase," Dillon explains.

With Erin's custom GUI containing an array of buttons and functions, AuraLock certainly ended up looking and feeling professional. "We feel that all features included in the app are key to a positive user experience," she says. And it definitely put a smile on our faces.

Quick FACTS

- The system needs to be powered 24/7
- It unlocks when it recognises approved faces
- AuraLock can also be controlled via an app
- It uses the processing power of Raspberry Pi 4
- OpenCV has more than 2500 algorithms

Erin originally wanted to create an iOS app, but decided on Android because of the availability of helpful resources at the University of Wyoming



PicoKenbak

Recreating one of the first personal computers with a microcontroller shows how far we've come, as **Rob Zwetsloot** finds out

MAKER

Kosmas Raptis

A high school student from Greece who works with computers and electronics in his spare time, and enjoys low-level and emulator-based projects.

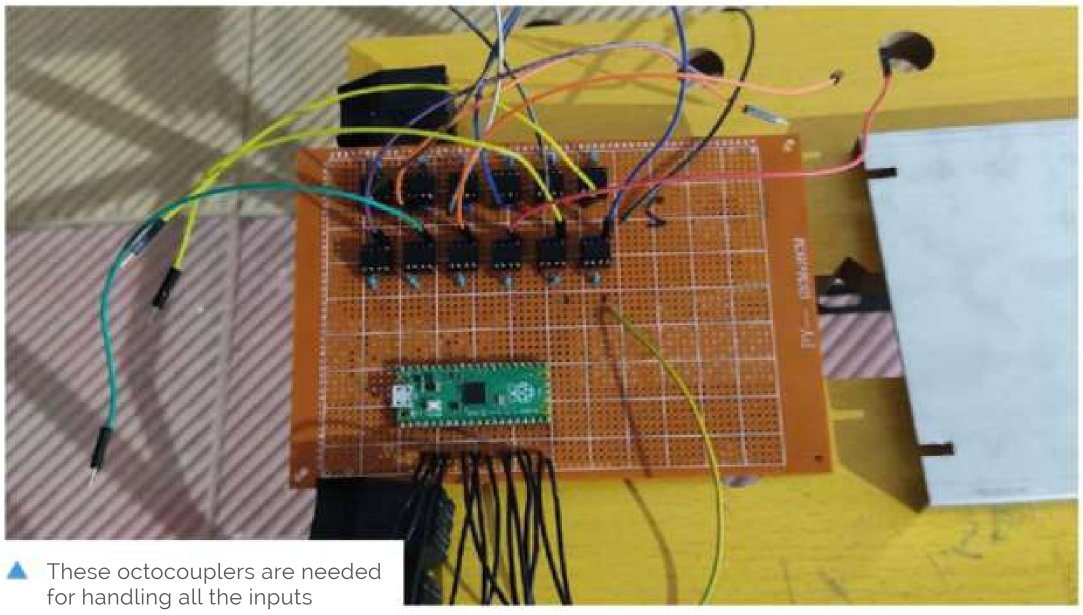
magpi.cc/kosmasgit

A few issues ago we featured a recreation of the Kenbak-1 using a Raspberry Pi 4 in 2:5 scale – it's considered to be the very first personal computer by many, and originally came out in 1971. Raspberry Pi 4 is definitely much more powerful thanks to Moore's Law and, it turns out, so are microcontrollers.

"[This] is something I call the PicoKenbak," Kosmas Raptis tells us via email. "It's a functional recreation of the Kenbak-1, one of the first (if not the first) personal computers, using a Raspberry Pi Pico. My version does not really look similar to the original computer due to how I was not able to find a similar case or similar push-buttons, but it functions the exact same way for the end user."

Kosmas is a fan of low-level programming and, when asked to write a paper on how computer programming has evolved over the decades, he decided to dig deep and go the extra mile.

"As I was researching about the beginnings of personal computers, I came across the Kenbak-1, and apparently this was a quite well-documented system, with the original creator having a website that explained how it was made and how it worked," Kosmas explains. "When I saw that, I immediately realised it would be really cool for my presentation. However, it is not easy to get one of these, and the recreation kits were not really viable for me either, so I decided to build my own."



These octocouplers are needed for handling all the inputs

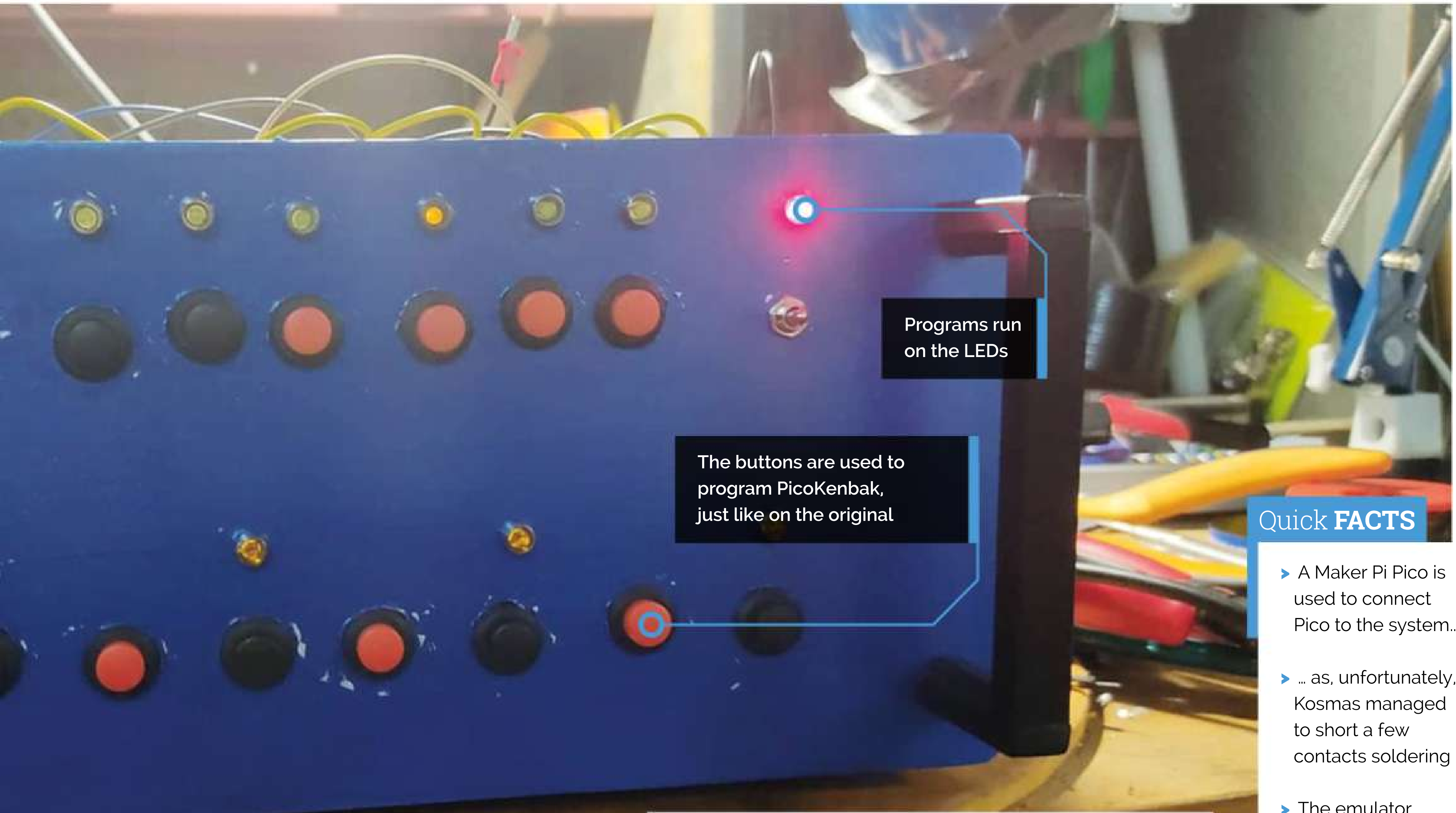
Kosmas had to improvise with a box, basically making it from scratch himself

More Pico

At the time of creation, Kosmas was interested in using Pico due to the low price, and because he wanted to see more cool Pico projects in the world.

"Actually building the project was more difficult than I expected," he says. "The case is basically a big metal box that I drilled some holes through and painted blue and was easy enough to create, but setting up the actual wiring was not easy. This project uses 15 push-buttons and 13 LEDs (one power LED connected to the power supply as an on/off indicator, and eight I/O and four function LEDs connected to the Pico), which all had to be confined to the small space of the case.

"Everything was also hand-soldered, so there were some issues with soldering something at this scale for someone as new to soldering as me... this gets me to how the project actually works. The Kenbak-1 (and, by extent, PicoKenbak) had twelve LEDs. Eight of the them were used for data I/O, and



Programs run on the LEDs

The buttons are used to program PicoKenbak, just like on the original

Quick FACTS

- A Maker Pi Pico is used to connect Pico to the system…
- … as, unfortunately, Kosmas managed to short a few contacts soldering
- The emulator currently runs as fast as Pico can…
- … which is faster than the Kenbak-1, but Kosmas wants to fix that
- Kosmas built the hardware first, then did the software

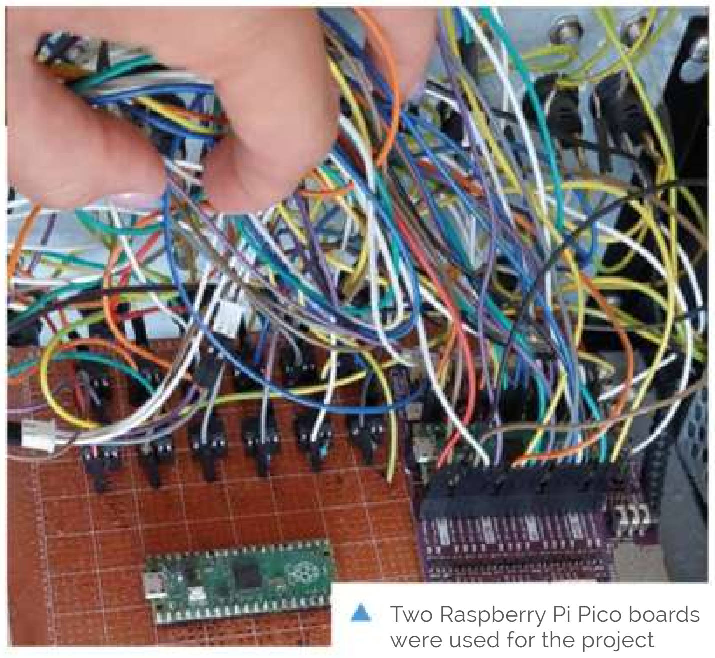
> Actually building the project was more difficult than I expected

four of them were used for indicating that various functions were done, like clearing input, showing the current memory address, reading from memory, and stopping program execution.

"Additionally, I had to use optocouplers to power the LEDs, to ease up power requirements on the Pico. There are also 15 push-buttons. Eight of those are used for inputting data to store into memory, and the other seven are used for functions. Clearing input (which I have wired to the Pico's RUN pin), set/show memory address, read/store with memory, and start/stop program execution. This was actually a close one because, by the time I was done, I had used up every pin on the Pico."

Emulation and more

Kosmas wrote an emulator in C, which handles all the original functions of the Kenbak-1.



▲ Two Raspberry Pi Pico boards were used for the project

"Programming is done the same way that it was done with the original Kenbak-1," he explains. "The programmer uses the push-buttons to type an instruction byte in binary, and then stores the byte into memory. Program space can start at address 4 and end at address 127. By default, execution starts from address 4."

In the future, Kosmas plans to update it with a Pico W and add some wireless functionality, such as executing code over a network. If you want to check out the emulator, the code is available online at **magpi.cc/picokenbak**.

SUBSCRIBE TODAY FOR JUST £10

Get 3 issues + FREE Pico W

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
RETRO GAMING WITH PICO W
THE TOP 10 RASPBERRY PI & PICO CASES
54 PAGES OF
BACK TO SCHOOL

+

Subscriber Benefits

- FREE Delivery
  Get it fast and for FREE
- Exclusive Offers
  Great gifts, offers, and discounts
- Great Savings
  Save up to 35% compared to stores

Subscribe for £10

- Free Pico W
- 3 issues of The MagPi
- Free delivery to your door
- £10 (UK only)

Subscribe for 12 Months

- Free Pico W
- 12 issues of The MagPi
- Free delivery to your door

£55 (UK)	£90 (USA)
£80 (EU)	£90 (Rest of World)

☎ Subscribe by phone: **01293 312193**

Subscribe online: **magpi.cc/subscribe**

Email: **magpi@subscriptionhelpline.co.uk**

Subscribe for £10 is a UK-only offer. The subscription will renew at £30 every six months unless cancelled. A free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. (No Raspberry Pi Pico W with £5 Rolling Monthly Subscription).

SUBSCRIBE TODAY AND GET A FREE Raspberry Pi Pico W

Subscribe in print today and get a **FREE development board**

WORTH $6

- A brand new RP2040-based Raspberry Pi Pico W development board
- Learn to code with electronics and build your own projects
- Make your own home automation projects, handheld consoles, tiny robots, and much, much more

This is a limited offer. Not included with renewals. Offer subject to change or withdrawal at any time.

Buy now: magpi.cc/subscribe

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
LEARN ELECTRONICS
WITH PICO W
54 PAGES OF PROJECTS & TUTORIALS

SUBSCRIBE on app stores

From **£2.29**

Available on the App Store

GET IT ON Google Play

SUCCESS STORY magpi.cc/success

Directed Machines

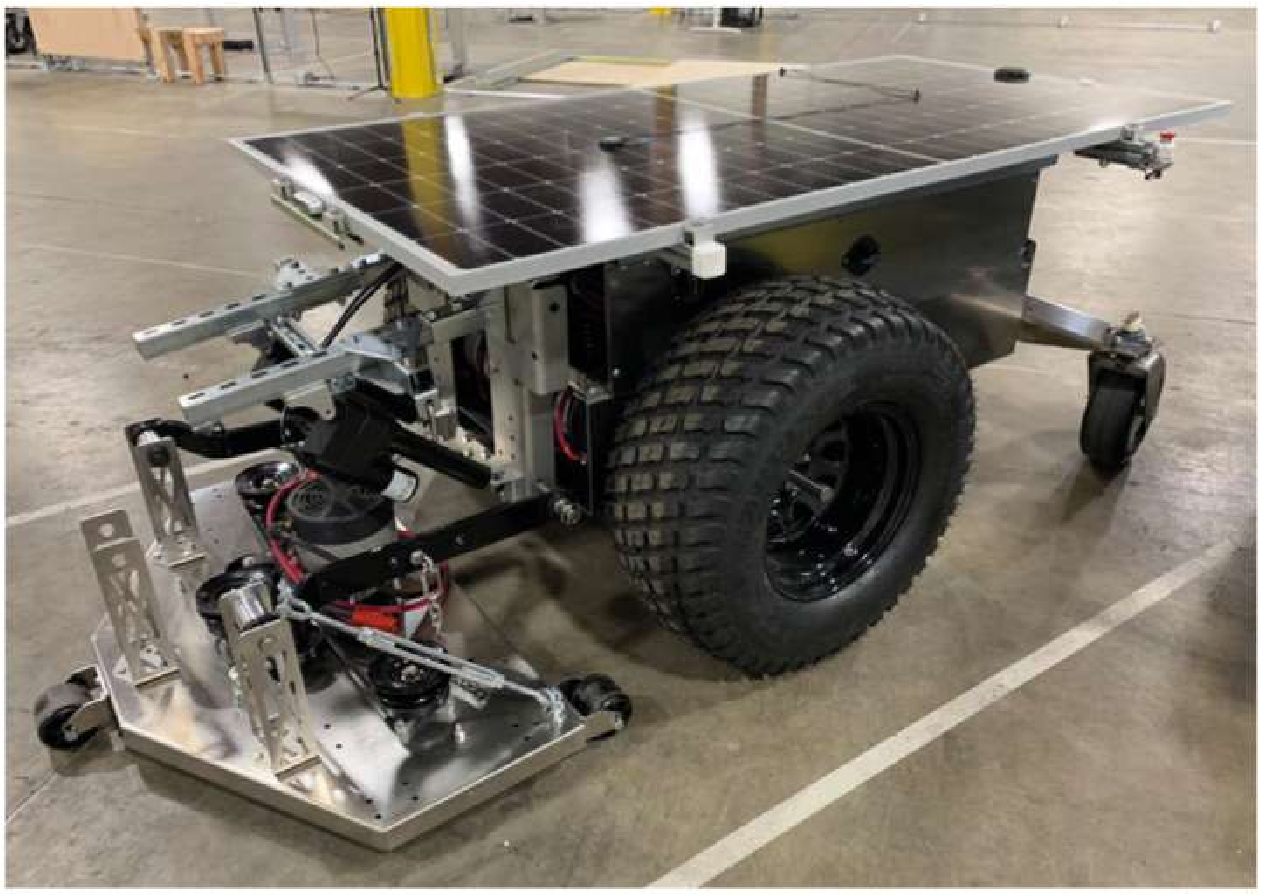
Directed Machines' Raspberry Pi-based Land Care Robots fulfil precision agricultural tasks and are both customisable and repairable, reports **Rosie Hattersley**

With the global population steadily rising, optimising agricultural productivity and land use to grow enough food is an ever more pressing issue. Raising crops is a precision industry, and one in which farmers are constantly under pressure to produce more from their acreage in order to meet the demand for food, at a cost to keep their main customers, the supermarkets, competitive. Minimising the use of pesticides and fertilisers is critical to maintaining profits, as well as preventing soil degradation, while labour to sow and harvest crops is in increasingly short supply. Seasonal labour shortages can be eased by drafting in casual workers from other states or countries, but the bigger trend is towards the increased mechanisation and industrialisation of farming.

Directed Machines (**directedmachines.com**) is a small but important player in the agri-robots sector, developing and marketing Land Care Robots (LCRs) for a wide range of vertical markets. Raspberry Pi 4 is a crucial aspect of the company's successful designs (**magpi.cc/robotfarmers**).

THE CHALLENGE

When George Chrysanthakopoulos stepped away from a prestigious 21-year career in software development, he was keen to capitalise on his thorough knowledge of robotics: he ran a 1200-engineer robotics department at Microsoft before moving to VMware, where he became a fellow. For his own, post-industry, project he began designing solar-powered robots to help deal with the annual snowfall. Discussions with neighbouring small-scale farmers quickly convinced George there were myriad possible uses for robots in agriculture, and there was a healthy market for low-cost mechanised assistants that could help solve some of their pesticide and plant management. Profit margins on small farms are especially tight and farmers would need to be able to perform multiple tasks and to repair their robot on-site, meaning readily available parts and a fairly simple but robust hardware was imperative. Nonetheless, there seemed to be a market for relatively low-cost, heavy-duty, solar-electric, autonomous machines.



Robots could have multiple uses across the growing season – preparing the ground, applying fertiliser and pesticides, sowing seeds, removing weeds, transporting seedlings from place to place ready for planting out, and harvesting the crops. Each of these tasks would need to be performed accurately in order to justify the robot's outlay and be a true replacement for hard to find seasonal manpower. The right to repair is a big issue in the agricultural industry, so there were important considerations regarding maintenance and the servicing of complex machinery.

WHY RASPBERRY PI?

George founded dCentralized Systems and set about developing a modular robot that could be used for a variety of agricultural tasks. Prototypes of the first Directed Machines-branded product, a laser-guided autonomous weeding robot, had field trials in early 2018 and were followed by the more powerful Land Care Robot – the base unit for Directed Machines' modular offering.

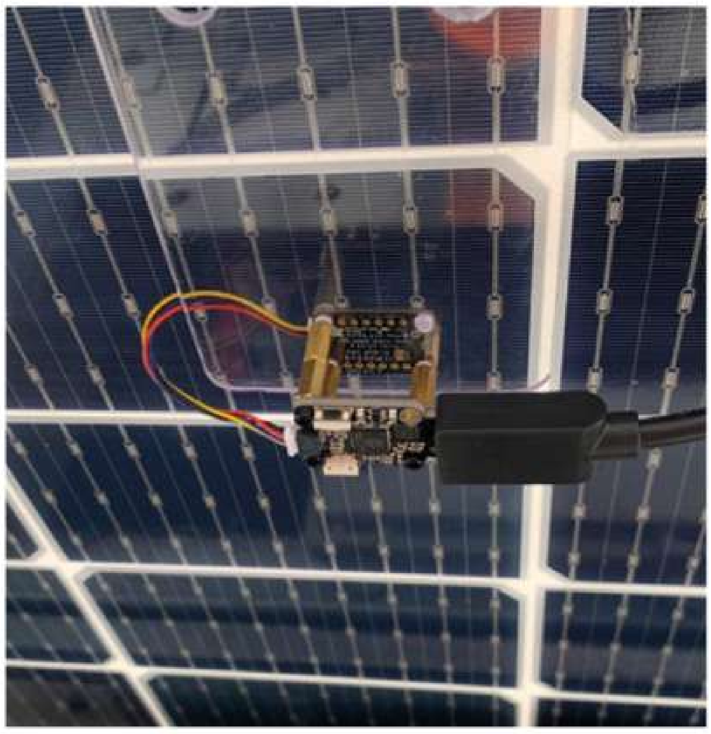
> You can attach different implements to our robot

The Land Care Robot was built around Raspberry Pi for cost reasons, as well as the reliability and availability of hardware. The company reasoned that with a huge installed base and active community, the chances of bugs being found in Raspberry Pi modules was minimal. This, of course, is critical when it comes to machinery that may be remotely located and would therefore need to be updated or replaced by customers. "If you can do it on a Raspberry Pi, there are huge advantages to doing [so]," says Dan Abramson, Directed Machines' co-founder and COO. "You've sold millions and millions of units, the chances of bugs popping up are really, really low." He also mentions the "richness of the ecosystem around the Raspberry Pi" and the relatively low energy requirements. "So if you can do it on a Raspberry Pi, there are compelling reasons to do it on a Pi."

Dan adds: "We like 3iC architecture, and Raspberry Pi is a very beautiful match for our first principles, algorithmic architecture philosophy. Instead of throwing brute-force computing power at solving engineering problems, we leverage the structure inside those problems to reduce the computational requirements related to those problems. After imposing those structural constraints, and reducing the scope of the problem, Raspberry Pi is a beautiful tool, a beautiful partner to execute the calculations that we need to execute."

THE RESULT

Directed Machines has been selling across North America in constrained fashion for the past two years, meaning plenty of advance research to ensure Land Care Robots are a suitable fit for each client's needs. Development was entirely in-house and took just two years including mechanical development, electrical development, and all the software development. For the past two years, Directed Machines has been selling into market and deploying the Land Care Robot platform "in all sorts of different interesting use cases and verticals," says Dan. "It's an autonomous robot, powered by a file, so you can attach different



implements to our robot to allow it to perform different tasks. You might attach a mower deck to our robot to allow it to mow grass. You might attach a snowplough to our robot to allow it [to plough snow], you might attach a cart to our robot to allow it to pull whatever it is you want to be pulled in.

"With certain agricultural customers, as their crop comes to harvest, the crop is attacked by birds and other predators seeking food. We've attached a sky puppet, like you would see at a car wash, to the back of our robot, and the robot moves around fields or the orchard autonomously, carrying the sky puppet scaring away the birds.

"Other customers include golf courses, solar farms, rural properties, and nurseries that have to tow potted plants in very heavy volume. The LCR can be deployed in a wide range of terrain settings, from very, very flat properties that we need to perform tasks on, to one we have in Southern California that overlooks the Pacific Ocean, where the terrain is 55 degree slopes. The Raspberry Pi-powered LCRs are now also working full-time at utility scale on solar farms in the U.S. and have been warmly welcomed in Europe and Oceania too."

As well as handling very challenging terrains, the Land Care Robots are fully autonomous and are capable of completing multiple tasks without user invention. "We keep adding features to what we do all the time, and we continue to improve the robustness of our engineering and to experiment with new sensors. We are doing that more and more, using sensors that we find in Raspberry Pi's ecosystem."

Working alongside a curated set of customers during field tests, and since, has worked well for both sides: clients get early access to new features while Directed Machines gets invaluable, directly applicable feedback. Britt Fletcher owns Mutiny Bay Blues in Washington, and is a Directed Machines investor. His organic blueberry farm "focuses on regenerative farming practices and keeping our carbon footprint as small as possible." Directed Machines approached Britt two years ago, keen to use Mutiny Bay a test site for its robotic solar-powered machine. "The Land Care Robot's ability to mow acres of a very structured row environment, as well as many of our open areas, while not using gas and also freeing up a farmworker to handle more difficult tasks, made coming on board an easy decision," says Britt. Working in a high-stakes structured environment can be a particular challenge and, at first, the accuracy was not where it is now, says Britt, but LCR updates developed in response to Britt and other customers' feedback means Mutiny Bay now has a machine that Britt trusts to work "totally unsupervised in the rows," as this video shows: **magpi.cc/automowing**.

"The cost is about three quarters of what it had been and, more importantly, frees up increasingly scarce human capital."

In 2021, Blueweave Consulting estimated the smart agricultural robot market to be worth approximately $11.5B, and predicts this to nearly double to $24.3B by 2028, citing demand for precision agriculture as a key driver (**magpi.cc/agrirobotsmarket**).

Daren Richins runs an 'entrepreneurial' farm in Utah. He believes the direction agriculture is headed is "fairly obvious", with labour shortages and increased capital expense taking a big toll on farmers. "I think the answer to these challenges lies within robotics and technology. As I've had the opportunity to work with my Land Care Robot, I've been able to peer into a not-so-distant reality that relies quite heavily on this type of technology to satisfy the challenges we face today in agriculture. Directed Machines has been an excellent partner in taking this bull by the horns."

RETRO GAMING with Raspberry Pi Pico and Pico W

What kind of video games can you run on a $6 microcontroller?

WRITER

K.G. Orphanides

K.G. is a software preservationist and developer with an abiding love of vintage computers.

magpi.cc/MightyOwlbear

From Dizzy to Doom, Raspberry Pi Pico and Pico W are holding their own against far more expensive emulation systems, and they're ready to play almost as soon as you power on. With the July 2022 launch of the Raspberry Pi Pico W, equipped with wireless networking capabilities, there are a few more features to be tapped.

Numerous mature and highly capable emulators have been ported to Pico, allowing you to run huge chunks of the 8-bit computer and console eras, with hardware kits and expansion boards that make it easy to connect monitors, controllers, SD card storage, and high-quality audio.

Unlike top-heavy computer emulators designed for computers and capable of running almost anything, if you spend enough time poking at the menus, RP2040 emulators become embedded systems, single-purpose, responsive, and fast-booting, just like the original hardware they're emulating.

Emulation is all well and good, but you can also use your Pico to bring brand new homebrew games to real retro hardware platforms, giving new life to beloved classic consoles. We've showcased two different flash carts, for the Nintendo Game Boy and N64 respectively, that use Pico with custom PCBs, that allow you to write your own retro console cartridges at a far lower cost than dedicated commercial flash carts.

▲ Play indie gems from modern C64 developers and publishers, like Psytronik's Honey Bee

DOOM
id SOFTWARE
PROVIDED BY id FREE OF CHARGE • SUGGESTED RETAIL PRICE $9.00 • id SOFTWARE, ©1993
ASUS
ELEGIANT

▲ Relive the gory glory days of the Doom, beloved by a generation of gamers

If you'd rather work with fresh hardware, we have the YouMakeTech's Pico 'GameBoy', a colour handheld console that wears its inspiration on its sleeve but is, in fact, designed to run and inspire the development of brand new games in MicroPython or C.

This little microcontroller can even play Doom, thanks to the gruelling porting and optimisation work invested by Raspberry Pi's in-house Pico SDK lead and classic games enthusiast Graham Sanderson, who's also responsible for the Pico's BBC Micro emulator (b-em) port.

Because this year marks the 50th anniversary of the first arcade release of Atari's Pong, on 29 November 1972, we've included the minimalist table tennis classic, showcasing a project that'll have you create a rainbow-colour handheld incarnation of the game, and demonstrating how to host a game of Pong on your local network by using Raspberry Pi Pico as a tiny web server. We're sure Pico W's future holds even more ambitious networked gaming applications.

> “We're sure Pico W's future holds even more ambitious networked gaming applications”

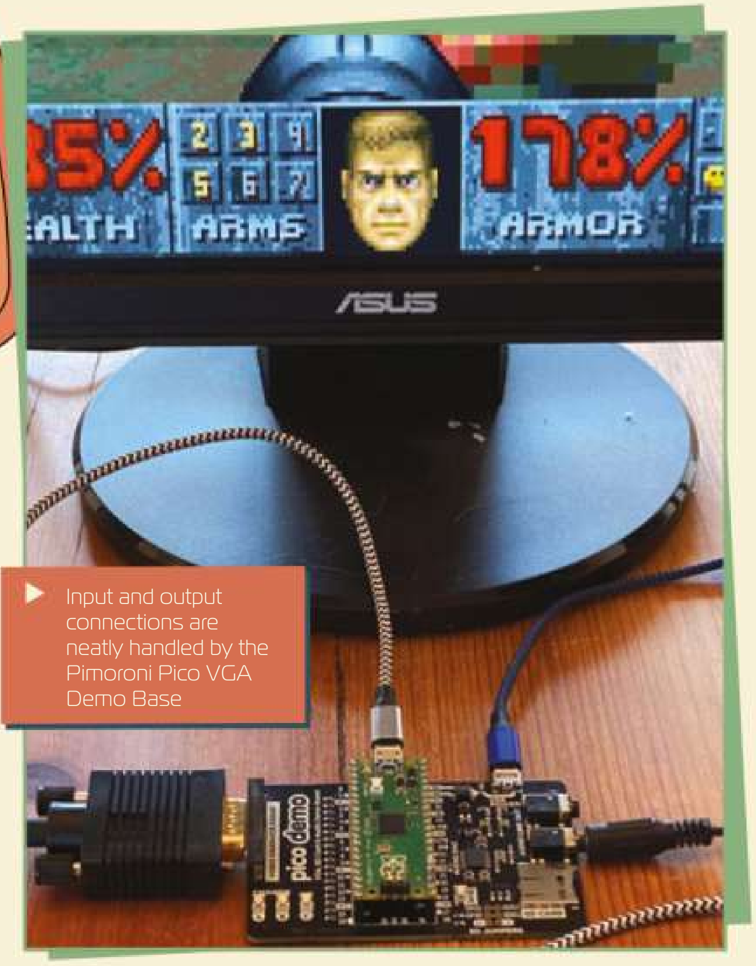
Join us on a whirlwind tour of some of the coolest Pico gaming projects. Whether you're interested in hacking hardware, developing software, or just hooking up fully functional gaming experiences on the most cost-effective hardware around, there's something here to Pico your interest.

Play Doom on Pico

We talk to Raspberry Pi Pico SDK lead Graham Sanderson about accurately porting Doom in all its glory to RP2040

Graham Sanderson, Raspberry Pi Pico SDK lead and performance architect by day, took on the challenge of making Doom run on the microcontroller in 2021. The game would be released in March 2022 after six months of development, but the first public evidence of the scheme appeared a year earlier, when Sanderson tweeted: "RP2040 Doom must be a thing, but if I do it, it needs to run all the demo WAD properly."

Sanderson grew up with the Sinclair ZX81 and BBC Micro. He says that the most fun part of his role working on the RP2040 SDK has perhaps been, "taking 30-plus years of development and applying it to the constrained microcontroller environment, which has a similar feeling set of constraints to the home computers I had as a child."


Input and output connections are neatly handled by the Pimoroni Pico VGA Demo Base

He based RP2040 Doom on the Chocolate Doom (**chocolate-doom.org**) port of the original source code, itself designed to remove some of the limitations associated with the original DOS version.

The greatest challenge, he says, was getting the entire 4MB demo to fit, along with the game itself, into the Pico's 2MB flash storage. "The original game data file from Doom (DOOM1.WAD) is over 4MB big, and I have included everything: all the graphics, levels, music, sound, splash screens, multiplayer networking – the lot."

Particularly impressive is a tool called whd_gen, part of the RP2040 Doom codebase, that converts and compresses WAD files into a custom format that reduces the level files' size by up to 57% in the case of DOOM1.WAD. WHD, which stands for 'Where's Half the Data?', is particularly effective at reducing Doom's graphical overheads, detailed in Sanderson's behind-the-scenes article on the techniques used to create this port.

Pico porting projects

Doom is by no means Sanderson's first venture into porting classic games to the RP2040: "The very first thing I ported was actually a ZX Spectrum emulator. It is on my list to open-source; it is just in a bit of a mess. This one was actually developed while we were still developing the RP2040 on FPGA and, so, was limited to a

Alert!
Copyright

Video game files are protected by copyright law. Be sure to use ROM files that have been released with the owner's blessing, or modern homebrew games designed to be shared. There are lots of legal options.

magpi.cc/legalroms

system clock of 48MHz. It also, at one point, ran with less memory."

The ZX Spectrum emulator wasn't cycle accurate, but this didn't really matter as far as the performance of most Spectrum games is concerned. But that doesn't apply to every retro emulation system.

"Given the new-found freedom of a higher clock rate," – the production version of Raspberry Pi Pico has a maximum clock speed of 133MHz – "I thought I'd have a go at porting a BBC emulator where it is critical that everything happens at exactly the right cycle."

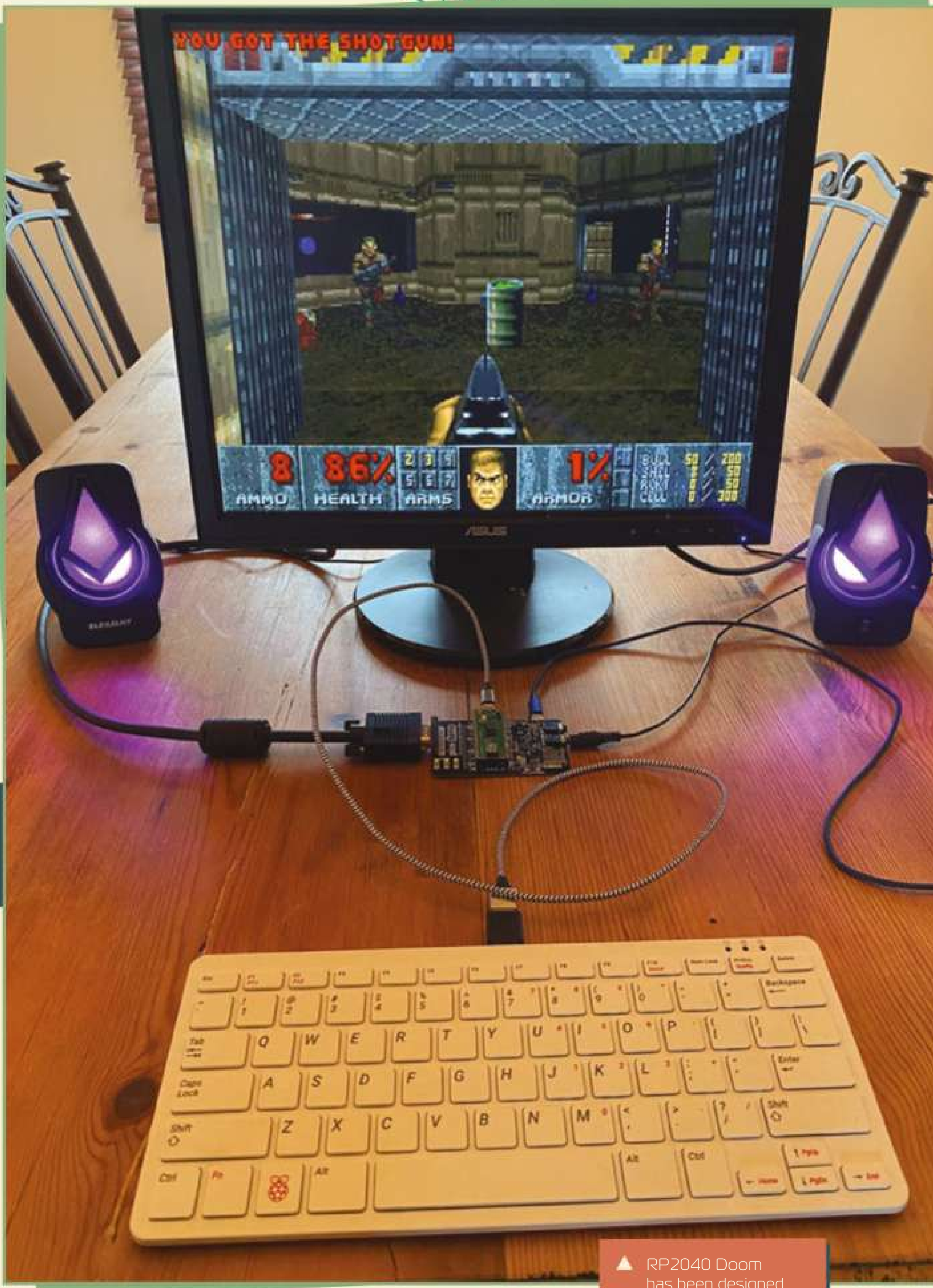
Sanderson says he does it for the challenge, to see if it's possible. That's a sentiment you'll hear from a lot of the coders and hardware hackers we've spoken to for this feature.

"I usually start with another code base, and of course they are never designed to run on something this constrained. [The] first thing is generally to divorce them from the idea that look-up tables many hundreds of kB big are a good idea, and [I] generally have to rewrite large portions with some sort of new methodology [or] approach and redo certain bits in assembly language."

"He does it for the challenge"

RP2040 Doom shows the huge potential of one small microcontroller, a great deal of patience and determination, and three decades' worth of experience.

But you don't need to start with that level of experience to begin working on your own Raspberry Pi Pico projects. And if you prefer using other people's work as a platform for your own, higher-level projects, or as the basis of a really cool make, there's a wealth of kits and code to enjoy playing with.



▲ RP2040 Doom has been designed to be faithful to the DOS original in graphics, sound, and responsiveness

Download your own Doom

You can download the release binary of RP2040 Doom at **magpi.cc/rp2040doom**, enter BOOTSEL mode, and copy the UF2 file you want over to it. To run it, Pico will have to be connected to a vgaboard-compliant RP2040 graphics board. The larger, 4MB file is designed for other RP2040-based devices, with more flash storage.

It's designed to run with a Pimoroni Pico VGA Demo Base (**magpi.cc/vgademobase**), which is functionally identical to the "VGA, SD card, and audio demo board" described in Raspberry Pi's 'Hardware Design with RP2040' documentation (**magpi.cc/rp2040hardware**). Sanderson notes, "It is a lot easier to use the VGA Demo Base, though!"

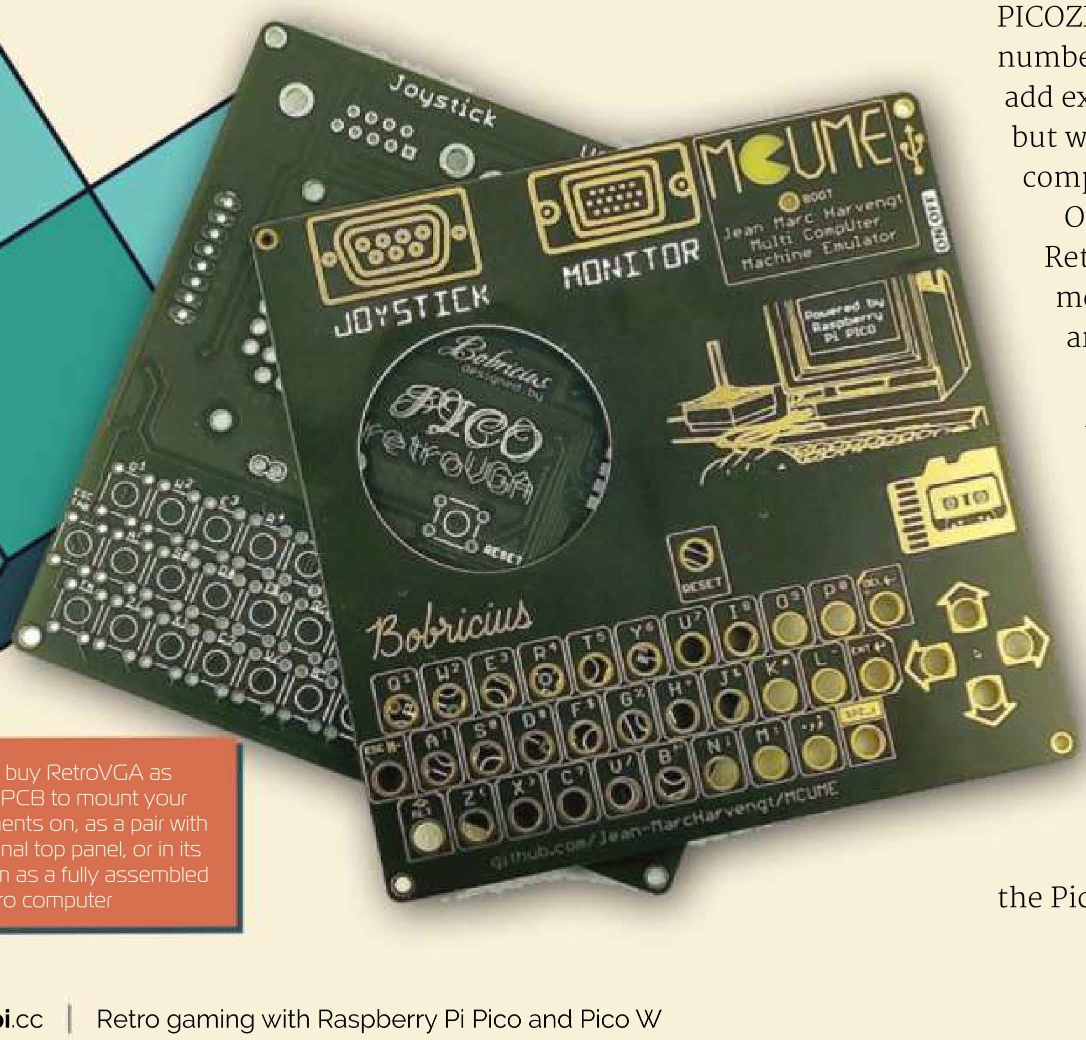
Although that's the easiest way to hook up display, controls, and sound, it's not the only way. In some of Sanderson's YouTube videos of the project, you'll see a breadboarded version of the setup, with the video GPIOs hooked up to a VGA connector via a resistor DAC and the audio ones to an PCM5102 I2S DAC board.

You can read a full account of Graham Sanderson's feat of optimisation at **magpi.cc/makingrp2040doom**.

Hardware showcase: RetroVGA

A fully-fledged retro computer in your pocket, with DB9 joystick input and VGA output

One of the most impressive Pico project kits around is Peter 'Bobricius' Misenko's RetroVGA, a Raspberry Pi Pico multi-retro computer which equips the microcontroller with a VGA output, DB9 joystick port, QWERTY keyboard, 3.5 mm audio output, integrated piezo speaker, USB port for input devices on supported emulators, and an SD card slot for storage.



▲ You can buy RetroVGA as a single PCB to mount your components on, as a pair with an optional top panel, or in its final form as a fully assembled multi retro computer

The whole thing measures just 10 cm square, small enough to fit in a coat pocket, just in case you need to carry a retro microcomputer around with you. If you use a DB9 joystick, you'll need to reduce its cable length to around 30 cm, or it'll register false key presses.

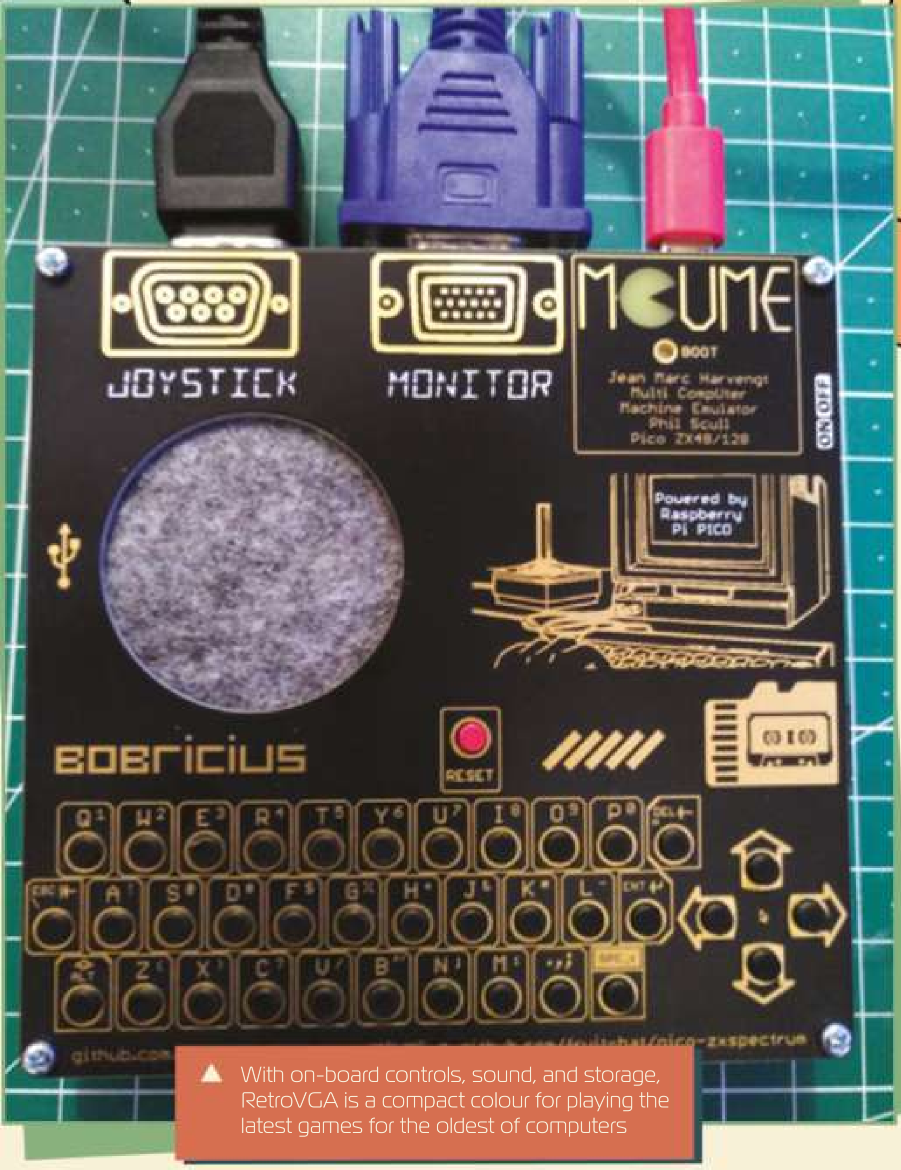
The keyboard isn't quite full, lacking F-keys that are needed in some C64 titles, for example. The RetroVGA's sister device, the PICOZX (**magpi.cc/picozx128**) changes the number of GPIO pins devoted to VGA output to add extra keys, if that's a deal-breaker for you, but with a limited number of GPIO connections, compromises have to be made somewhere.

Once you've selected your game from the RetroVGA's SD card storage, the keyboard means you can LOAD and RUN it, and there are even directional keys to play with.

RetroVGA was first built to work with MCUME, Jean-Marc Harvengt's Multi CompUter Machine Emulator, and Miroslav Nemecek's PicoVGA display library. It now also supports Phil Scull's more feature-packed pico-zxspectrum emulator.

PicoVGA graphics standard

Harvengt and Nemecek were closely involved in the development. The RetroVGA board's video output wiring is compliant with the PicoVGA standard, and not with the Pico SDK's default vgaboard standard.



▲ With on-board controls, sound, and storage, RetroVGA is a compact colour for playing the latest games for the oldest of computers

It uses different GPIO to VGA connections, which are documented in the project repository at **magpi.cc/picovga**.

Full documentation for RetroVGA itself is available at **magpi.cc/retrovgadocs** to help you both assemble RetroVGA and get it up and running with a selection of emulators or the PicoVGA library, and there's enough detail there to get developers up and running

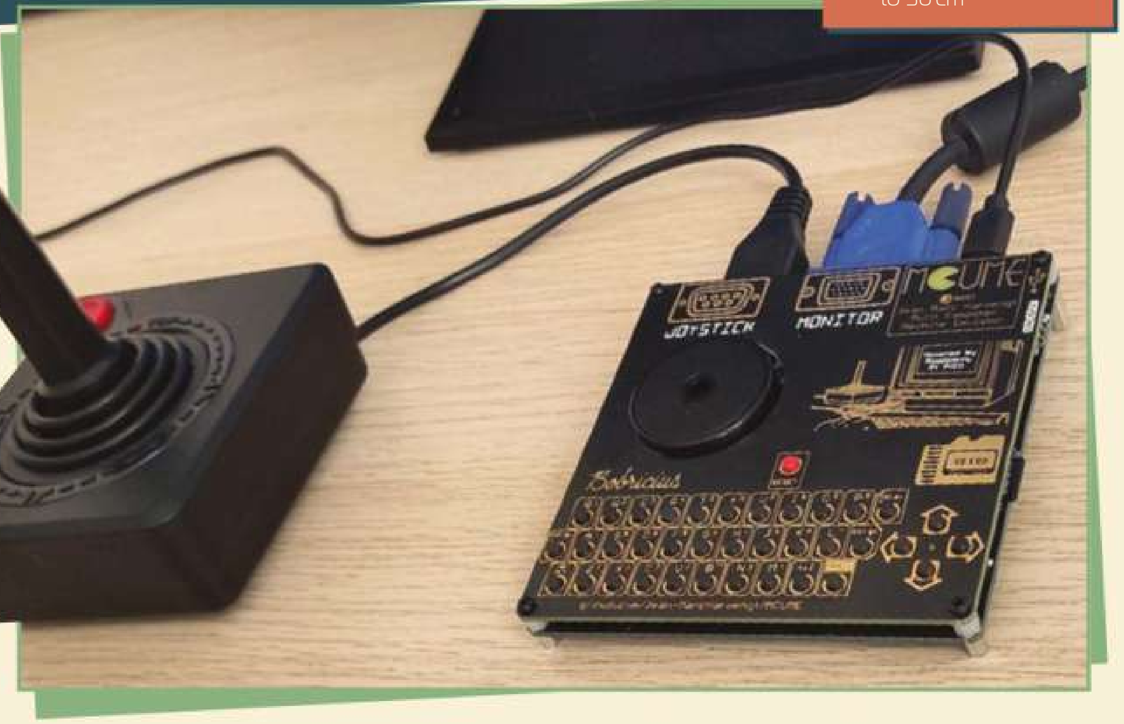
> “There are plenty of options available, depending on your needs and on how much soldering you're prepared to do”

Currently, you can use MCUME to emulate the ZX81, ZX Spectrum, Atari 800, C64, VIC20, Atari 2600, Odyssey/Videopac, Colecovision, and Atari 5200. MCUME provides an effective, but bare-bones emulation experience. It can read game files from the SD card, giving you access to the rich world of C64 and Spectrum homebrew and indie releases. The pico-zxspectrum emulator is even more capable, supporting quicksave states, as well as USB keyboards and joypads.

Even when using a multi-device emulator such as MCUME, you'll use one emulator at a time. MCUME binaries are available to download at **magpi.cc/picoretrovga** and the UF2 files for pico-zxspectrum all live at **magpi.cc/zxuf2**.

While some of the projects in this overview of Pico-based retro gaming are still in their experimental stages, calling for breadboard assemblages or requiring you to have PCBs custom-made, you can just buy a RetroVGA kit at **magpi.cc/retrovga**.

Starting at $10 for the bare main PCB, $30 for a PCB and gold keyboard-bearing top panel, up to $108 for a fully assembled, ready-to-flash unit with all components, there are plenty of options available, depending on your needs and on how much soldering you're prepared to do.



▼ For best input results, you'll need to reduce the length of your DB9 joystick's cable to 30 cm

Hardware hacking

Four retro gaming hardware projects to test out

PICOCART64

The PicoCart64 is a flash cart for the Nintendo N64: a device that can hold code to be read by the original console hardware, allowing you to load your own games and software onto an actual N64.

It can boot N64 homebrew, which makes it a compelling way of accessing the console's rich post-market game ecosystem on real hardware, given that the most popular N64 flash carts are both costly and often hard to come by.

Creator Konrad Beckmann says that he set out on the project to find out if it's possible to use Raspberry Pi Pico instead of the FPGAs (field-programmable gate arrays) used in most commercially available flash carts, already expensive hardware that's become even more rare in the face of global chip shortages.

The PicoCart64 Lite is the project's first functional prototype flash cart, capable of loading N64 games, test files, and homebrew. It requires a single Raspberry Pi Pico, plus a MOSFET and resistor, and costs less than $10 to make.

The project's GitHub repository includes component lists, PCB schematics and, helpfully for those new to having their own PCBs fabricated, a link to a comparatively inexpensive service that'll do just that.

You'll find both hardware schematics and software in the develop branch of the PicoCart repository, and active discussion among contributors and hardware hackers implementing the tech on the Dubious Technology Discord channel.

Beckmann says that the PicoCart64 is "lowering the barrier, making it possible for people to make their own games and run it on actual hardware." Find N64 homebrew titles and SDKs to help you make your own at **magpi.cc/n64homebrew**.

magpi.cc/picocart64

PICO PONG

Software engineer Pip Austin has combined Raspberry Pi Pico with Pimoroni's rainbow LED Unicorn board (**magpi.cc/picounicorn**), a multicoloured 16×7 matrix of RGB LEDs with four integrated buttons, to create an eye-catchingly luminous game of Pong.

She chose Pong because, as she says, it's "a classic game with simple rules, but is visually very powerful." This aesthetic aspect is something that the Pico Unicorn really brought to life. Her own favourite parts of the game are visually arresting: the trailing light behind the ball, and the scrolling text when you win.

Released in 2021, Pico Pong was Austin's first venture into Python programming, and her self-written tutorial (**magpi.cc/picoponggame**) does a great job of showing her workings and explaining why she made each decision in her code, making it an inspiring resource for programmers new to the language.

But if you just want to get this shining, rainbow-bright Pong interpretation into your hands, you can find her complete MicroPython program in her GitHub repo, ready to copy to Pico with an IDE such as Thonny. She's hoping to make more games on her Pico in the future: "I have started work on Rat On A Scooter – making use of the scrolling function!"

magpi.cc/ponggit

PICO-GB-CART

If you'd like to channel the spirit of Nintendo's classic handheld a bit more directly, you can always turn your Pico into a fully-fledged Game Boy flash cart. John Green was inspired by his experience making Game Boy emulators to see if he could make a hardware emulator with hardware skills he describes as "basic".

He used Gekkio's GB-BRK-CART (**magpi.cc/gbbrkcart**), a Game Boy breakout board, to interface with the handheld, learn what signals the Game Boy was expecting, and test to see if the Pico was fast enough to send I/O signals to it.

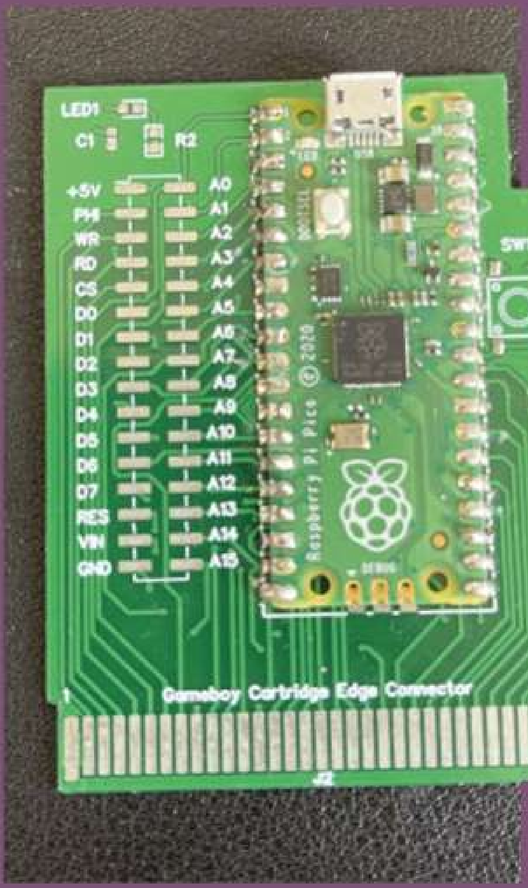
Then, he designed another prototype board that allowed for the Pico to be directly soldered to the PBC. It worked, but the way both these prototype boards interfaced with the Game Boy was a little unreliable, he says: "the Game Boy runs on 5V and the Pico runs at 3.3V, but after testing various voltage level converters, etc., I could never come up with a way of reading/writing data back and forth from the Game Boy and Pico whilst keeping the 'correct' voltages."

The make was fairly challenging: "You would think with the Game Boy only running at 4MHz and Pico being able to run at 133MHz base that [timing] should be no issue," Green says, but that didn't take into account time for the signal to be decoded by the Pico, get the next CPU instruction to send back to the Game Boy, and set the correct GPIO pins. In the end, Green found that 133MHz was too slow and Pico had to be overclocked to 360MHz. Only one of his ten Picos was able to function stably at this speed.

Green hasn't returned to a project's codebase for a while, but has plans for its future, including adding the ability to load game images from an SD card.

You can find instructions and software for the PICO-GB-CART at its GitHub repo, while schematics for the bespoke PCB created for the project by HDR can be found at **magpi.cc/rp2040gbcart**.

Green's other Pico projects include a fighting game hitbox controller, and you can find a overview of personal software projects, such as a Game Boy raycaster, alongside a glimpse at his professional work in the games industry at **0xen.github.io.**



magpi.cc/picogbcart

YOUMAKETECH PICO GAMEBOY

One of the most aesthetically pleasing Raspberry Pi Pico projects we've seen is this Raspberry Pi Pico GameBoy, a complete playable console with a custom 3D-printed case, created by Vincent of YouMakeTech. Vincent supplies the STL files you'll need to create the case, a detailed parts list, connection diagrams, and a video assembly guide.

Although it's designed to look like Nintendo's classic handheld, albeit at three quarters the size of its original inspiration, the Raspberry Pi Pico GameBoy is no simple emulation console.

Equipped with a 1.54 in, 240×240, 65K colour screen, a D-pad, and two buttons, it exists to encourage tinkerers and would-be developers to explore game programming in their choice MicroPython or C++.

It's one of a series of Pico-based 3D-printed consoles that Vincent has created. If you're after a pure emulation console, YouMakeTech's Pico-GB (**magpi.cc/picogbemu**) can actually play DMG games designed for the original Game Boy (not the Game Boy Color) using the RP2040-GB emulator (**magpi.cc/rp2040gb**).

That includes the many modern homebrew Game Boy games that you can find on platforms such as Itch.io (**magpi.cc/gbgames**).

magpi.cc/picogb

Make Pico Pong

Host HTML games, including Pong, on Pico W and play them over your local network

Top Tip

Last known address

Most DHCP routers will reassign Pico the IP address it had last time; to save checking for it, assign it a static IP address.

Most of the projects we've looked at in this feature will work perfectly with a standard Raspberry Pi Pico, but it's about time we did something that can take advantage of Pico W's unique networking capabilities. Let's host some web games.

There are limits to what you can host. Complex Twine games throw memory errors, for example. But as we're interested in retro games, we're in luck. To help celebrate Pong's 50th anniversary, here's a complete beginner's guide to hosting a two-player, HTML5 and JavaScript version of Pong on Pico, from installing MicroPython

01 Install Thonny

If you're using any Raspberry Pi computer to write to Pico, the Thonny Python IDE should already be installed. You'll find it in the repositories of other Linux distributions, while Windows and macOS users can download it from **thonny.org**.

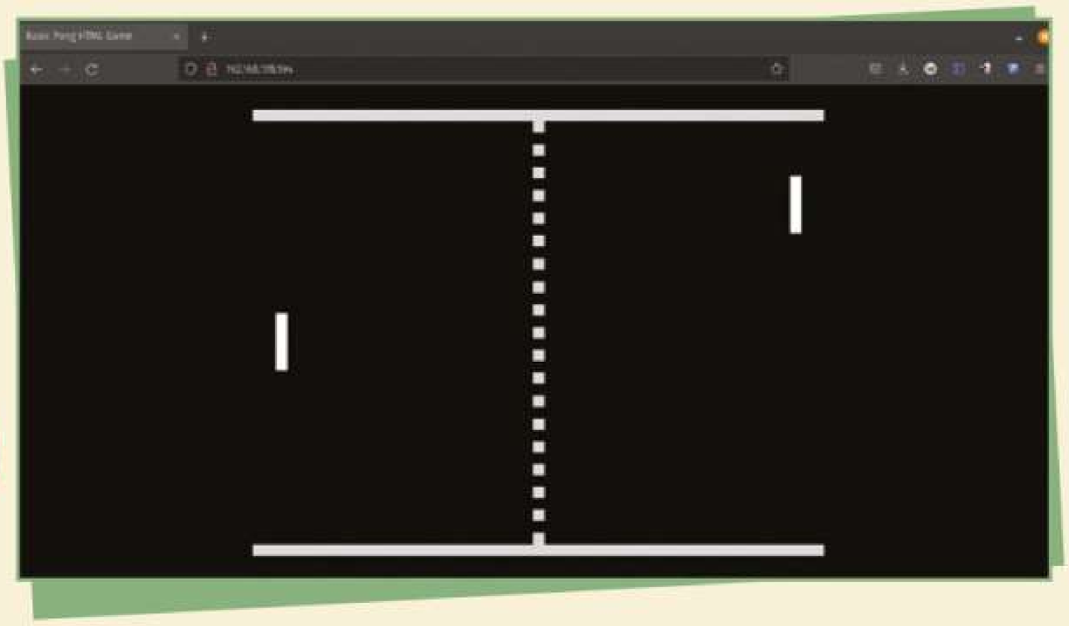
Nuke it from orbit

If you start encountering memory errors, or need to get rid of files from previous projects or experiments, you can use the `os.remove` command to delete files one at a time from the REPL interpreter prompt. But if you need to comprehensively clear everything, MicroPython included, there's a more dramatic option.

Download **flash_nuke.uf2** from **magpi.cc/flashnuke** (direct file download). Unplug Pico (if connected), hold down the BOOTSEL button, and plug it in to your PC.

Copy **flash_nuke.uf2** over to it, and it'll reboot.

Don't forget to copy a fresh MicroPython nightly build over to it before embarking on your next project.



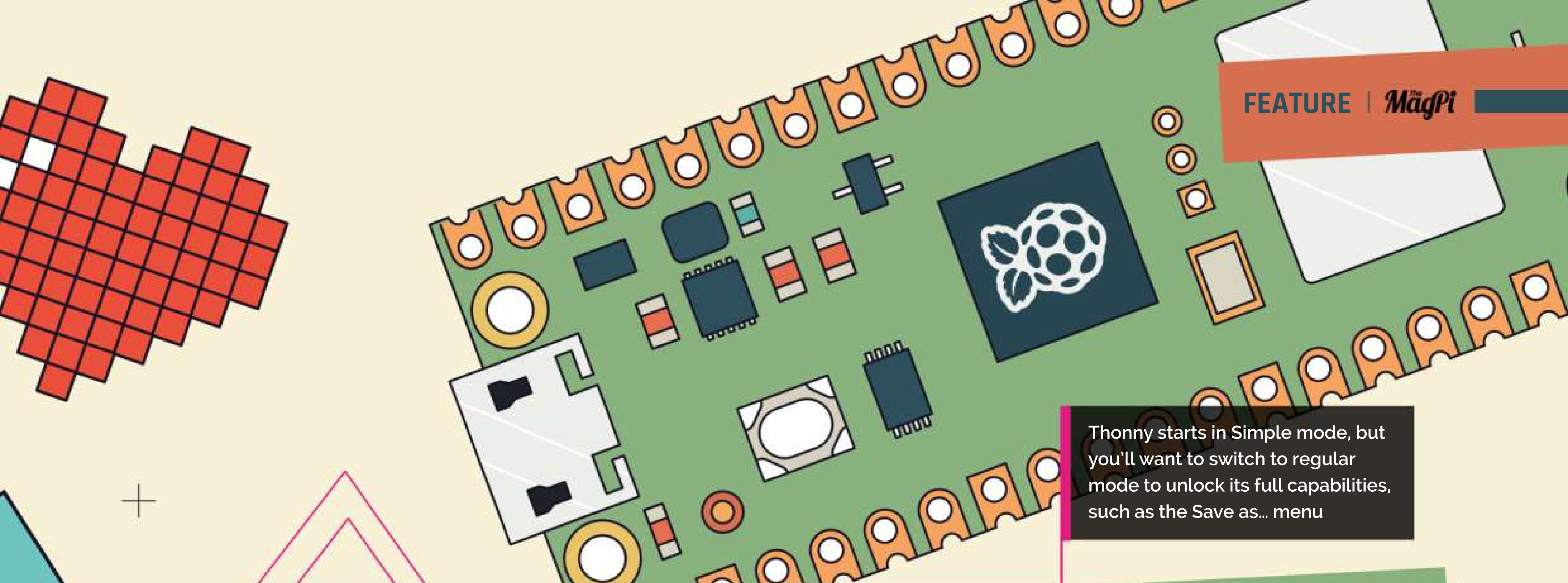
▼ A two-player game of Pong is served over your local network

02 Install MicroPython

Grab a fresh nightly build of MicroPython from **magpi.cc/rp2picow**. Make sure Pico W is unplugged, hold down its BOOTSEL button, and simultaneously connect it to your PC. It'll appear as a mass storage device. Copy the UF2 file you downloaded over to it. Pico W will reboot.

03 Prepare Thonny

Open Thonny. In the bottom right of its window, you'll see a line of text indicating what



interpreter you're using. If this does not currently say 'MicroPython (Raspberry Pi Pico)', click on the text line and select that option.

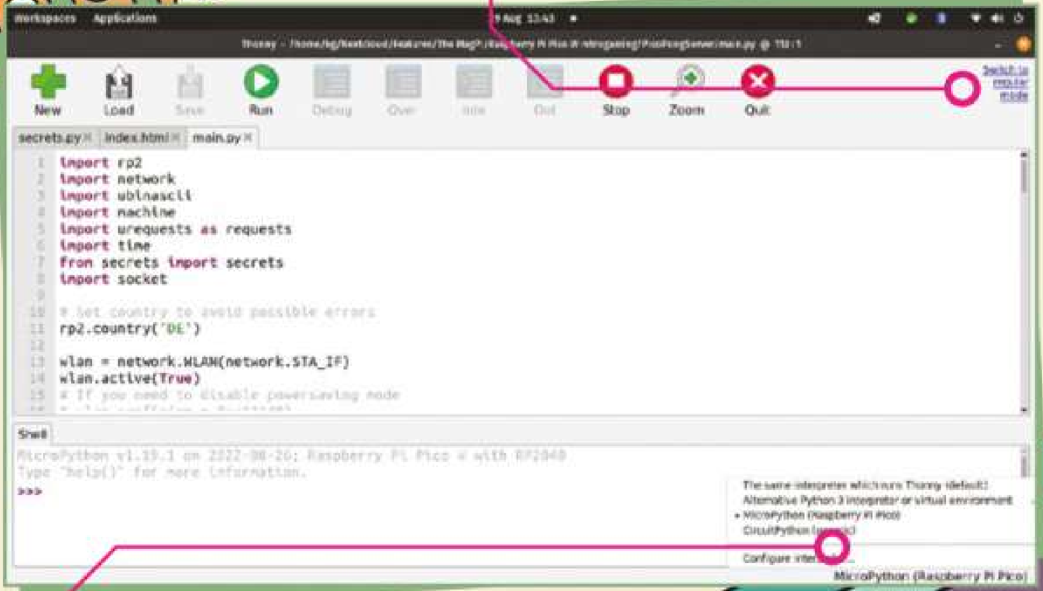
04 Scripting a server

We're going to use two scripts to get our Pico W intranet game server online: **main.py** is your server code, and **secrets.py** holds the SSID and password for your wireless network, allowing Pico W to automatically connect and start serving its content whenever it's powered on in vicinity of the network's access point. Our scripts are based on MIT-licensed project server scripts (**magpi.cc/picowledserver**) created by Nathan Bustler of Pi Cockpit. Download our versions of both scripts from our project page at **magpi.cc/picopongserver**.

> "We're going to use two scripts to get our Pico W intranet game server online"

05 Customise your network config

In Thonny, open our **main.py** and **secrets.py** scripts, update **secrets.py** with your own WiFi network's SSID and password, then save them to your connected Pico W. This is easiest if your copy of Thonny is in Regular mode rather than the Simple mode it starts in. If you're in Simple mode, click the 'Switch to regular mode' link at the top right of Thonny's icon bar. In Regular mode, use the File menu's 'Save as...' option.


Workspaces
Applications
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
secrets.py
index.html
main.py
Shell
The same interpreter which runs Thonny (default)
Alternative Python 3 interpreter or virtual environment
• MicroPython (Raspberry Pi Pico)
CircuitPython (generic)
Configure interpreter...
MicroPython (Raspberry Pi Pico)

```
import rp2
import network
import ubinascii
import machine
import urequests as requests
import time
from secrets import secrets
import socket

# Set country to avoid possible errors
rp2.country('DE')

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
# If you need to disable powersaving mode
```

```
MicroPython v1.19.1 on 2022-08-26; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>
```

Thonny starts in Simple mode, but you'll want to switch to regular mode to unlock its full capabilities, such as the Save as... menu

Clicking on the text on the bottom right opens your interpreter selector. Make sure 'MicroPython (Raspberry Pi Pico)' is selected

06 Just add Pong

We're going to use Straker's CC-licensed Basic Pong Game (**magpi.cc/basicpong**). Download our mirror of the HTML from **magpi.cc/github**, open it in Thonny and save the file to Pico. You can customise this, for example by changing the `ballSpeed` variable to make the ball move slowly.

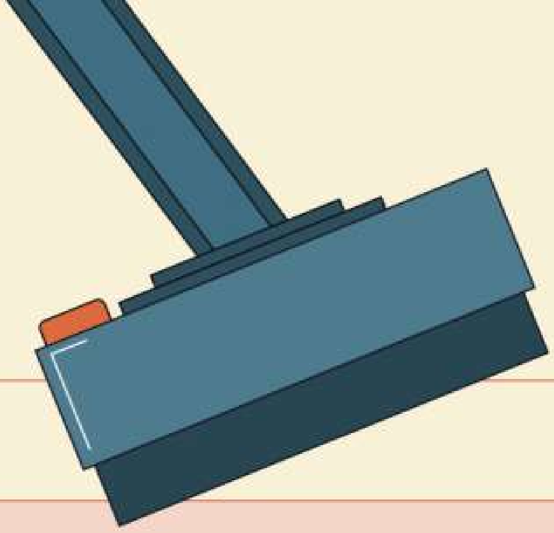
07 Serve your balls!

In Thonny, select the **main.py** file that you saved to Pico, and press the play button. Assuming your networking has been correctly configured, you'll see lines displaying the device's MAC and IP addresses. Copy the IP address into the browser of a computer connected to the same local network. Congratulations, you and a friend (or your right and left hands) can now play a classic Pong clone in your browser.

Top Tip

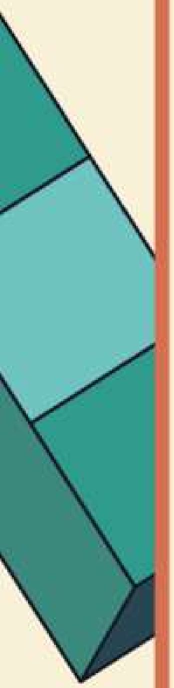
Further experiments

Try other lightweight HTML5 games and see how far you can push the Pico. It's also a great way to provide digital props and materials for physical games.

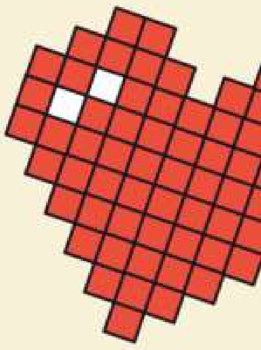


index.html

> Language: **HTML**

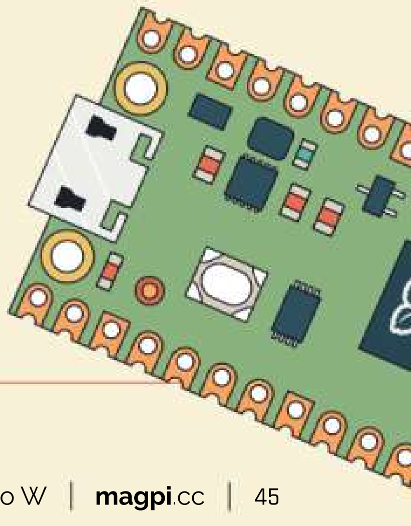


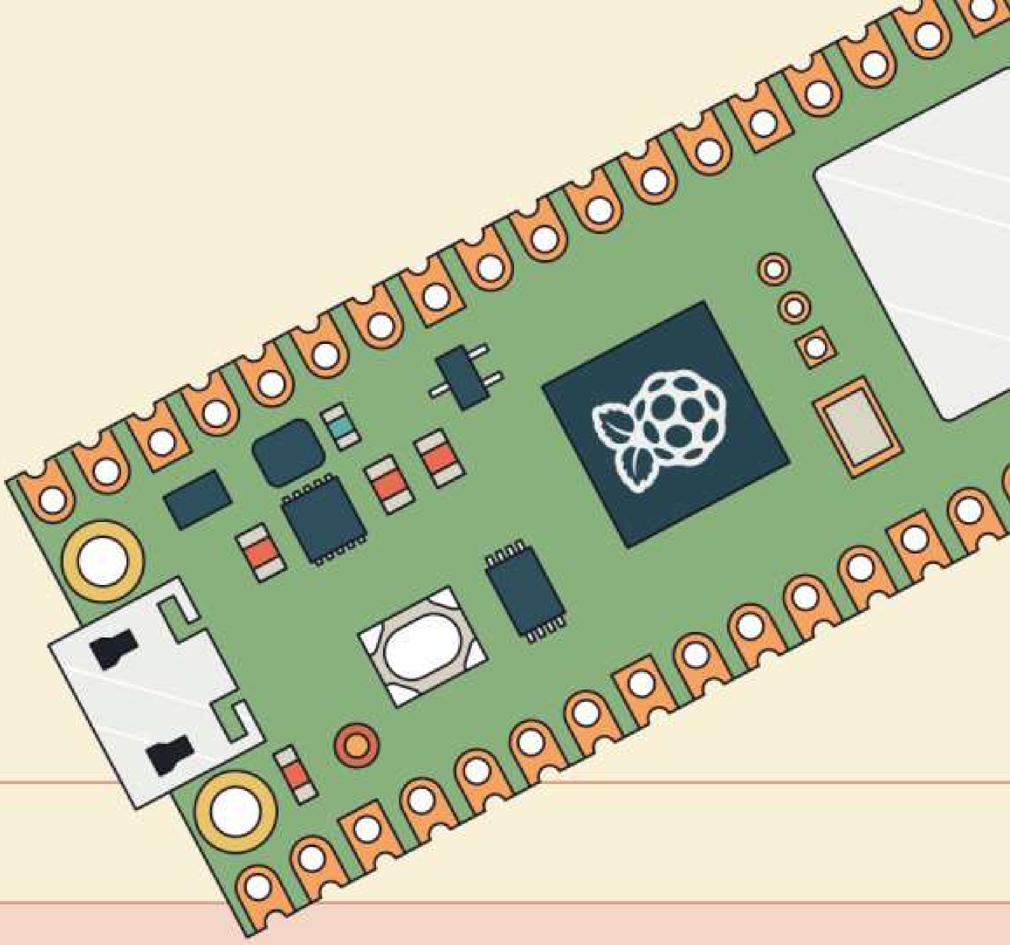
```
<!DOCTYPE html>
<html>
<head>
  <title>Basic Pong HTML Game</title>
  <meta charset=”UTF-8”>
  <style>
  html, body {
    height: 100%;
    margin: 0;
  }

  body {
    background: black;
    display: flex;
    align-items: center;
    justify-content: center;
  }
  </style>
</head>
<body>
<canvas width=”750” height=”585” id=”game”></canvas>
<script>
const canvas = document.getElementById(‘game’);
const context = canvas.getContext(‘2d’);
const grid = 15;
const paddleHeight = grid * 5; // 80
const maxPaddleY = canvas.height - grid - paddleHeight;

var paddleSpeed = 6;
var ballSpeed = 5;

const leftPaddle = {
  // start in the middle of the game on the left side
  x: grid * 2,
  y: canvas.height / 2 - paddleHeight / 2,
  width: grid,
  height: paddleHeight,

  // paddle velocity
  dy: 0
};
const rightPaddle = {
  // start in the middle of the game on the right side
  x: canvas.width - grid * 3,
  y: canvas.height / 2 - paddleHeight / 2,
  width: grid,
  height: paddleHeight,

  // paddle velocity
  dy: 0
};
const ball = {
  // start in the middle of the game
  x: canvas.width / 2,
  y: canvas.height / 2,
  width: grid,
  height: grid,

  // keep track of when need to reset the ball position
  resetting: false,

  // ball velocity (start going to the top-right corner)
  dx: ballSpeed,
  dy: -ballSpeed
};

// check for collision between two objects using axis-aligned bounding box (AABB)
// @see https://developer.mozilla.org/en-US/docs/Games/Techniques/2D_collision_detection
function collides(obj1, obj2) {
  return obj1.x < obj2.x + obj2.width &&
         obj1.x + obj1.width > obj2.x &&
         obj1.y < obj2.y + obj2.height &&
         obj1.y + obj1.height > obj2.y;
}

// game loop
function loop() {
  requestAnimationFrame(loop);
  context.clearRect(0,0,canvas.width,canvas.height);

  // move paddles by their velocity
  leftPaddle.y += leftPaddle.dy;
  rightPaddle.y += rightPaddle.dy;

  // prevent paddles from going through walls
  if (leftPaddle.y < grid) {
    leftPaddle.y = grid;
  }
  else if (leftPaddle.y > maxPaddleY) {
    leftPaddle.y = maxPaddleY;
  }

  if (rightPaddle.y < grid) {
    rightPaddle.y = grid;
  }
  else if (rightPaddle.y > maxPaddleY) {
    rightPaddle.y = maxPaddleY;
  }

  // draw paddles
  context.fillStyle = ‘white’;
  context.fillRect(leftPaddle.x, leftPaddle.y, leftPaddle.width, leftPaddle.height);
```



DOWNLOAD THE FULL CODE:
magpi.cc/github

```
  context.fillRect(rightPaddle.x, rightPaddle.y, rightPaddle.width, rightPaddle.height);

  // move ball by its velocity
  ball.x += ball.dx;
  ball.y += ball.dy;

  // prevent ball from going through walls by changing its velocity
  if (ball.y < grid) {
    ball.y = grid;
    ball.dy *= -1;
  }
  else if (ball.y + grid > canvas.height - grid) {
    ball.y = canvas.height - grid * 2;
    ball.dy *= -1;
  }

  // reset ball if it goes past paddle (but only if we haven’t already done so)
  if ( (ball.x < 0 || ball.x > canvas.width) && !ball.resetting) {
    ball.resetting = true;

    // give some time for the player to recover before launching the ball again
    setTimeout(() => {
      ball.resetting = false;
      ball.x = canvas.width / 2;
      ball.y = canvas.height / 2;
    }, 400);
  }

  // check to see if ball collides with paddle. if they do change x velocity
  if (collides(ball, leftPaddle)) {
    ball.dx *= -1;

    // move ball next to the paddle otherwise the collision will happen again
    // in the next frame
    ball.x = leftPaddle.x + leftPaddle.width;
  }
  else if (collides(ball, rightPaddle)) {
    ball.dx *= -1;

    // move ball next to the paddle otherwise the collision will happen again
    // in the next frame
    ball.x = rightPaddle.x - ball.width;
  }

  // draw ball
  context.fillRect(ball.x, ball.y, ball.width, ball.height);

  // draw walls
  context.fillStyle = ‘lightgrey’;
  context.fillRect(0, 0, canvas.width, grid);
  context.fillRect(0, canvas.height - grid, canvas.width, canvas.height);

  // draw dotted line down the middle
  for (let i = grid; i < canvas.height - grid; i += grid * 2) {
    context.fillRect(canvas.width / 2 - grid / 2, i, grid, grid);
  }
}

// listen to keyboard events to move the paddles
document.addEventListener(‘keydown’, function(e) {

  // up arrow key
  if (e.which === 38) {
    rightPaddle.dy = -paddleSpeed;
  }
  // down arrow key
  else if (e.which === 40) {
    rightPaddle.dy = paddleSpeed;
  }

  // w key
  if (e.which === 87) {
    leftPaddle.dy = -paddleSpeed;
  }
  // a key
  else if (e.which === 83) {
    leftPaddle.dy = paddleSpeed;
  }
});

// listen to keyboard events to stop the paddle if key is released
document.addEventListener(‘keyup’, function(e) {
  if (e.which === 38 || e.which === 40) {
    rightPaddle.dy = 0;
  }

  if (e.which === 83 || e.which === 87) {
    leftPaddle.dy = 0;
  }
});

// start the game
requestAnimationFrame(loop);
</script>
</body>
</html>
```

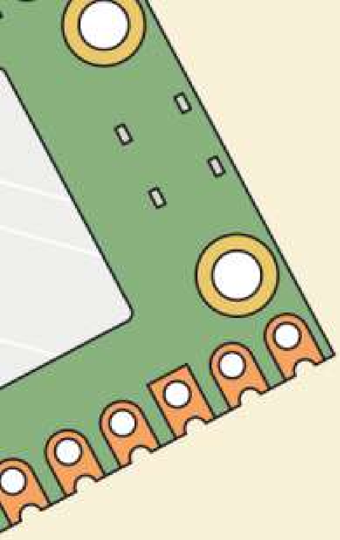


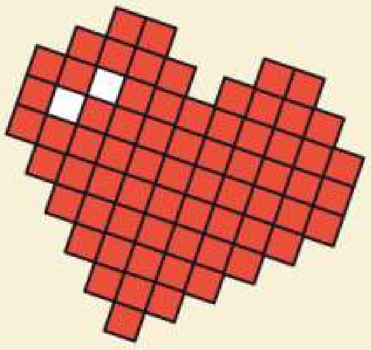


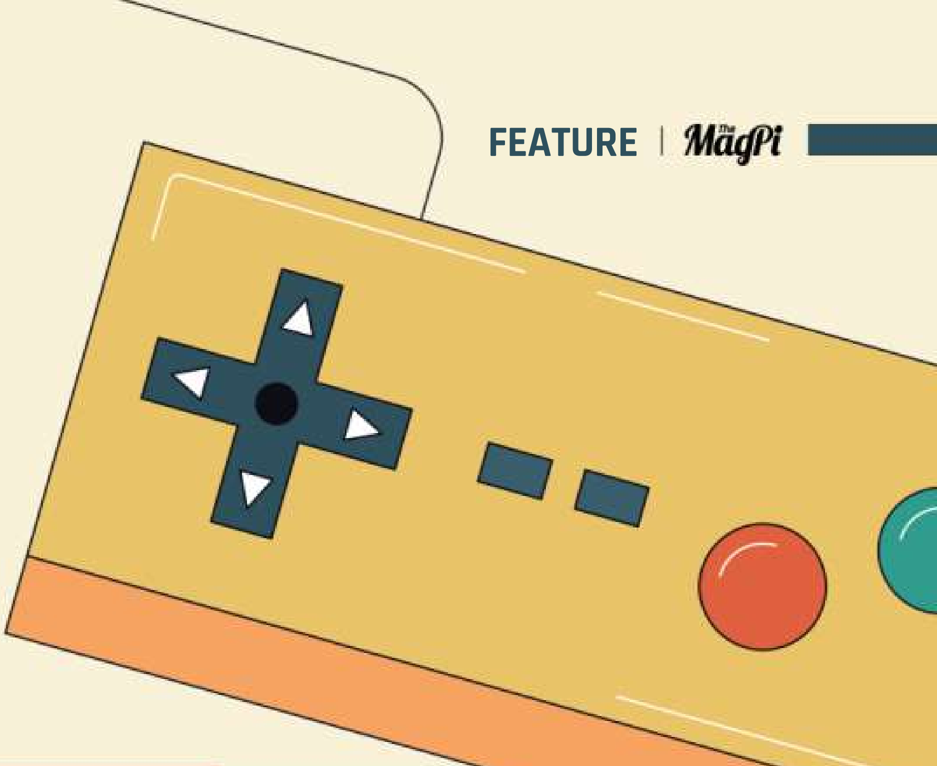
main.py

> Language: **Python**

```python
import rp2
import network
import ubinascii
import machine
import urequests as requests
import time
from secrets import secrets
import socket

# Set country to avoid possible errors
rp2.country(‘DE’)

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
# If you need to disable powersaving mode
# wlan.config(pm = 0xa11140)

# See the MAC address in the wireless chip OTP
mac = ubinascii.hexlify(network.WLAN().config(‘mac’),’:’).decode()
print(‘mac = ‘ + mac)

# Other things to query
# print(wlan.config(‘channel’))
# print(wlan.config(‘essid’))
# print(wlan.config(‘txpower’))

# Load login data from different file for safety reasons
ssid = secrets[‘ssid’]
pw = secrets[‘pw’]

wlan.connect(ssid, pw)

# Wait for connection with 10 second timeout
timeout = 10
while timeout > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    timeout -= 1
    print(‘Waiting for connection...’)
    time.sleep(1)

# Define blinking function for onboard LED to indicate error codes
def blink_onboard_led(num_blinks):
    led = machine.Pin(‘LED’, machine.Pin.OUT)
    for i in range(num_blinks):
        led.on()
        time.sleep(.2)
        led.off()
        time.sleep(.2)

# Handle connection error
# Error meanings
# 0  Link Down
# 1  Link Join
# 2  Link NoIp
# 3  Link Up
# -1 Link Fail
# -2 Link NoNet
# -3 Link BadAuth

wlan_status = wlan.status()
blink_onboard_led(wlan_status)

if wlan_status != 3:
    raise RuntimeError(‘Wi-Fi connection failed’)
else:
    print(‘Connected’)
    status = wlan.ifconfig()
    print(‘ip = ‘ + status[0])

# Function to load in html page
def get_html(html_name):
    with open(html_name, ‘r’) as file:
```







DOWNLOAD THE FULL CODE:

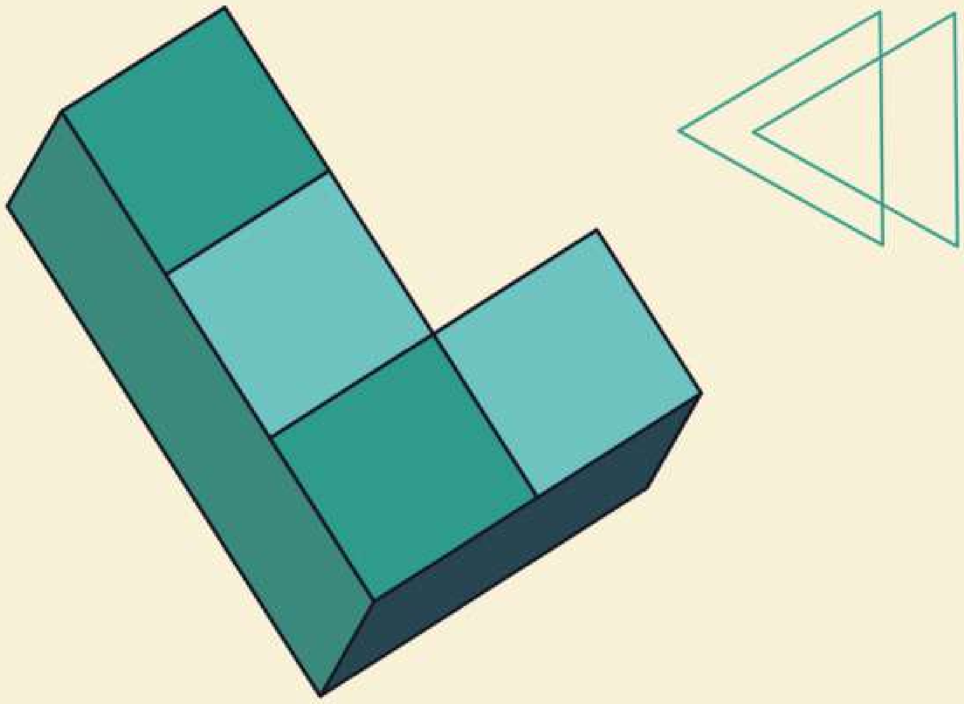
magpi.cc/github

```
        html = file.read()

    return html

# HTTP server with socket
addr = socket.getaddrinfo(‘0.0.0.0’, 80)[0][-1]

s = socket.socket()
s.bind(addr)
s.listen(1)

print(‘Listening on’, addr)
led = machine.Pin(‘LED’, machine.Pin.OUT)

# Listen for connections
while True:
    try:
        cl, addr = s.accept()
        print(‘Client connected from’, addr)
        r = cl.recv(1024)
        # print(r)


        response = get_html(‘index.html’)
        cl.send(‘HTTP/1.0 200 OK\r\nContent-type:text/html\r\n\r\n’)
        cl.send(response)
        cl.close()

    except OSError as e:
        cl.close()
        print(‘Connection closed’)

# Make GET request
#request = requests.get(‘http://www.google.com’)
#print(request.content)
#request.close()
```

secrets.py

> Language: **Python**

```
secrets = {
    ‘ssid’: ‘Enter_your_SSID_here’,
    ‘pw’: ‘Enter_your_Wi-Fi_password_here’,
    }
```



Part 01

Pico W IoT with Anvil: Mood lamp

Create a colour-changing RGB LED mood lamp controllable from a web app

MAKER

Phil King

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

You'll Need

- Raspberry Pi Pico W
- Anvil account (free tier) **anvil.works**
- RGB LED
- 3 × 330 Ω resistors
- 4 × male-male jumper wires

With built-in wireless connectivity, Pico W is ideal for creating Internet of Things (IoT) projects. To make it even simpler, Anvil has released a Pico W IoT toolkit to enable you to connect easily, and securely, to web apps you create on the Anvil platform. There's a drag-and-drop web UI builder, built-in database, user authentication, email integration, HTTP APIs, and more – and it all talks to your Pico W.

By using Anvil's special UF2 firmware image, you can code programs on Pico W in MicroPython as normal, with a few extra lines to connect to Anvil, then create a web app on the Anvil site which can send and receive data to/from your Pico W program. In this tutorial, we'll be showing you how to control the colour of an RGB LED using sliders in an Anvil web app.

01 Install the firmware

To be able to link your Pico W to the Anvil framework, you'll need to use a special firmware file. Go to **magpi.cc/aniluf2** and download the latest 'complete' UF2 file. (If you want to avoid overwriting any existing files on Pico W, use the 'firmware-only' version.)

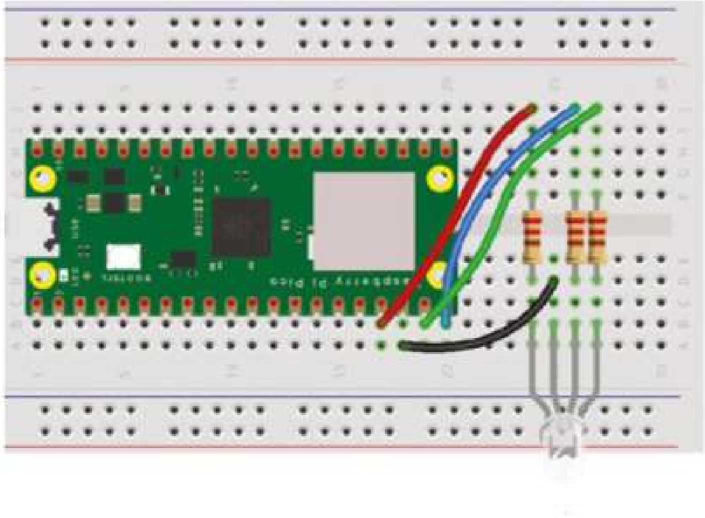


Figure 1: The wiring diagram for the mood lamp circuit using an RGB LED

As usual, hold Pico W's BOOTSEL button while connecting it to a computer via USB, then drag the UF2 file to the mounted 'RPI-RP2' volume. Once it's copied across, Pico W will automatically reboot and reappear as a volume called 'Pico W'.

02 Connect to WiFi

With Pico W connected to your computer, open up the Thonny IDE and make sure the Python interpreter (shown at the bottom right) is set to 'MicroPython (Raspberry Pi Pico)'. The **main.py** file on Pico W will run automatically, so you will need to stop it by pressing the Stop icon. Open the **boot.py** file and enter your wireless router's SSID (name) and password at the top.

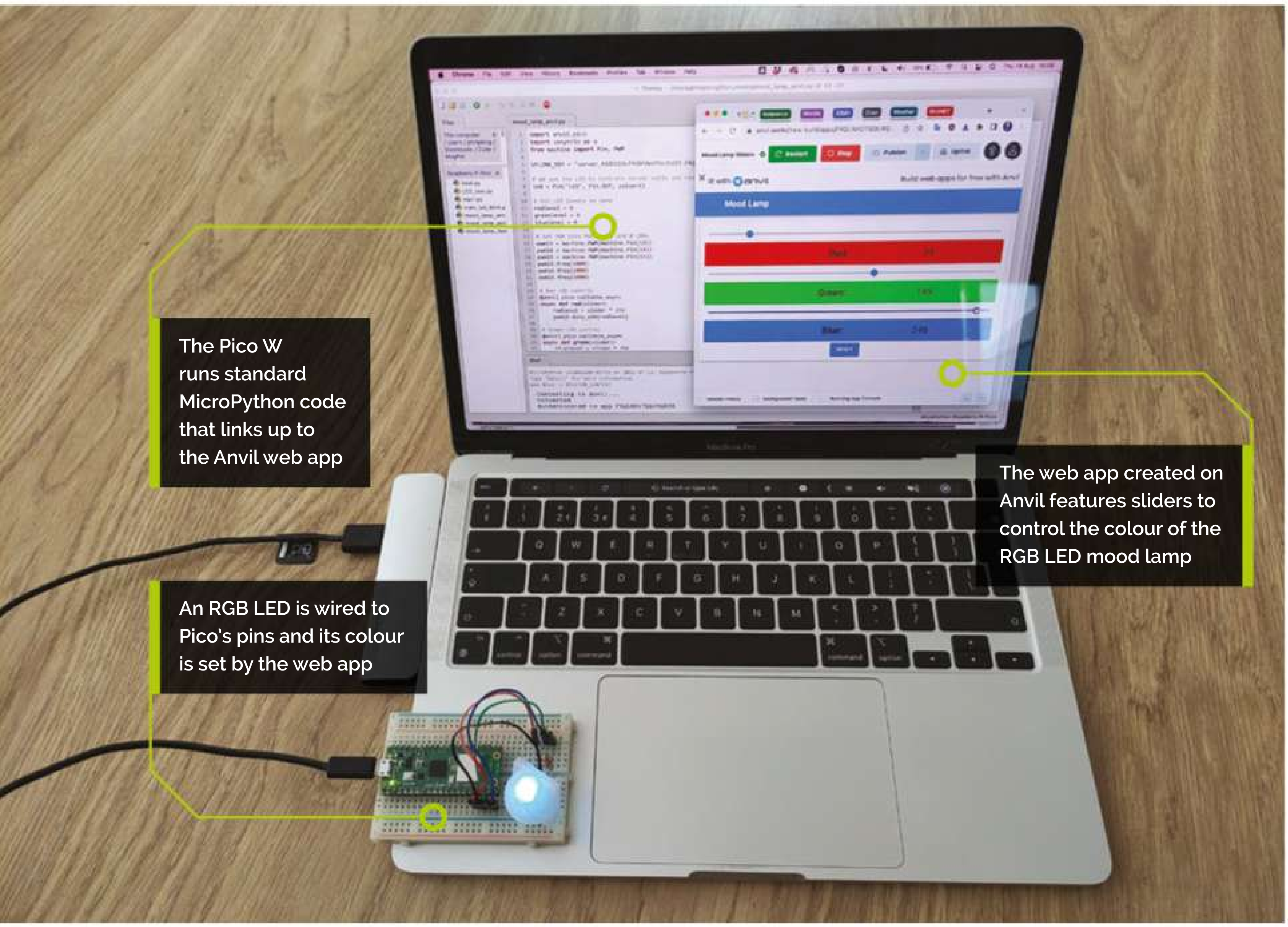
```
 WIFI_SSID = "<put your network name here>"
 WIFI_PASSWORD = "<put your wifi password
here>"
```

Now when Pico W reboots, it'll automatically connect to your wireless network.

03 Wire up the circuit

Place your Pico W on one end of a breadboard, as in **Figure 1**. Add your RGB LED at the other end with each leg placed in a different row. It has four legs, the longest of which should be the ground connection – use a jumper wire to connect that to a GND pin on Pico W.

As usual with LEDs, you should use resistors to limit the maximum current to avoid possible LED burnout. We've used three 330 Ω resistors placed over the central dip of the breadboard to connect the LED's R (red), G (green), and B (blue) legs to jumper wires connected to GPIO pins 13, 14, and 15 respectively on Pico W.



The Pico W runs standard MicroPython code that links up to the Anvil web app

The web app created on Anvil features sliders to control the colour of the RGB LED mood lamp

An RGB LED is wired to Pico's pins and its colour is set by the web app

Note: Our RGB LED's legs were in the order R, GND, B, and G, but yours may differ.

04 Test the circuit

To make sure everything is connected correctly, we'll run a simple program on Pico W. In the Thonny IDE on your computer, make sure the Python interpreter (shown at the bottom right) is set to 'MicroPython (Raspberry Pi Pico)'.

Open a new file and add the code from the **RGB_test.py** listing (overleaf). Run it and the RGB LED should turn red, green, and blue in turn. If the colours are in the wrong order, you will need to swap over the relevant jumper wires.

05 Write the code

Now let's write the MicroPython code for our RGB LED 'mood lamp', as seen in the **mood_lamp_sliders.py** listing. The top line, `import anvil.pico`, enables Pico W to connect to Anvil's servers. The second, `import uasyncio as a`, sets up an asynchronous scheduler for running concurrent functions. We also import the Pin and PWM classes from the machine library so we can control our RGB LED from the GPIO pins.

Further down is a line starting `UPLINK_KEY =`. This is where you will later need to paste the uplink key for the web app you create on Anvil so your Pico W program can link to it. At the bottom of the program is a line to connect using it.

> "Connect easily, and securely, to web apps you create on the Anvil platform"

Next, we create three variables to set the levels of the red, green, and blue parts of the RGB LED. We then set up the PWM pins as in our test code.

06 Call the decorator

In our code, we have three functions: one each for controlling the red, green, and blue parts of the RGB LED. Before each, we add a 'decorator' and also add `async` at the start of the line defining the function; for instance:

Top Tip

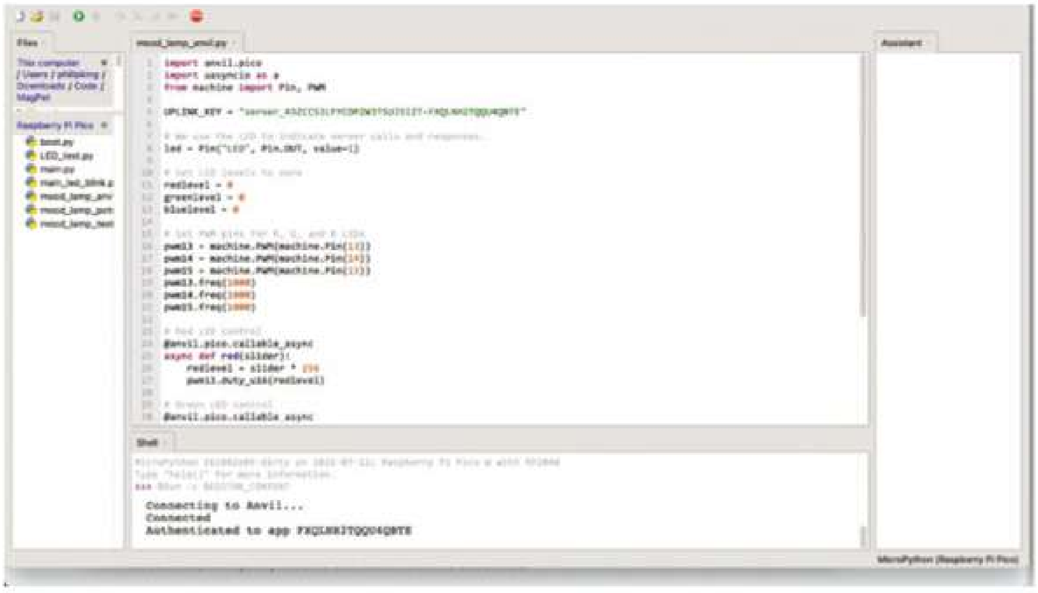
Zhuzh it up

Make your web app funkier by adding icons, images, or altering the colour scheme. We changed the slider text 'row_background' colours to red, green, and blue.

Top Tip

Autorun code

To make your Pico W code run automatically upon bootup, save it as **main.py**. If you already have a **main.py** file, you may want to save that under another name first.



```
import anvil.pico
import uasyncio as a
from machine import Pin, PWM

UPLINK_KEY = "server_[illegible]"

# We use the LED to indicate server calls and responses.
led = Pin("LED", Pin.OUT, value=1)

# Set LED levels to zero
redlevel = 0
greenlevel = 0
bluelevel = 0

# Set PWM pins for R, G, and B LEDs
pwm13 = machine.PWM(machine.Pin(13))
pwm14 = machine.PWM(machine.Pin(14))
pwm15 = machine.PWM(machine.Pin(15))
pwm13.freq(1000)
pwm14.freq(1000)
pwm15.freq(1000)

# Red LED control
@anvil.pico.callable_async
async def red(slider):
    redlevel = slider * 256
    pwm13.duty_u16(redlevel)

# Green LED control
@anvil.pico.callable_async
```

```
Connecting to Anvil...
Connected
Authenticated to app FXQLRKITQQU4QBTE
```

When run, the Pico W MicroPython code will connect to the linked Anvil web app

```
@anvil.pico.callable_async
async def red(slider):
```

This lets our connected Anvil web app know that this function is available to call from the web.

RGB_test.py

DOWNLOAD THE FULL CODE:

magpi.cc/github

> Language: **MicroPython**

```
from machine import Pin, PWM
from utime import sleep

# Set PWM pins to control R, G, and B LEDs
pwm13 = machine.PWM(machine.Pin(13))
pwm14 = machine.PWM(machine.Pin(14))
pwm15 = machine.PWM(machine.Pin(15))
pwm13.freq(1000)
pwm14.freq(1000)
pwm15.freq(1000)

# Loop to light R, G, B LEDs in turn
while True:
    # Red
    pwm13.duty_u16(65535)
    sleep(1)
    pwm13.duty_u16(0)
    sleep(1)
    # Green
    pwm14.duty_u16(65535)
    sleep(1)
    pwm14.duty_u16(0)
    sleep(1)
    # Blue
    pwm15.duty_u16(65535)
    sleep(1)
    pwm15.duty_u16(0)
    sleep(1)
```

We'll be sending a value to it, set by a slider in the web app, which will set the level of that colour. As we're using values from 0 to 255 for the slider, we multiply it by 256 to turn it into an unsigned 16-bit integer (0 to 65535) for the PWM level setting.

07 Design the web app

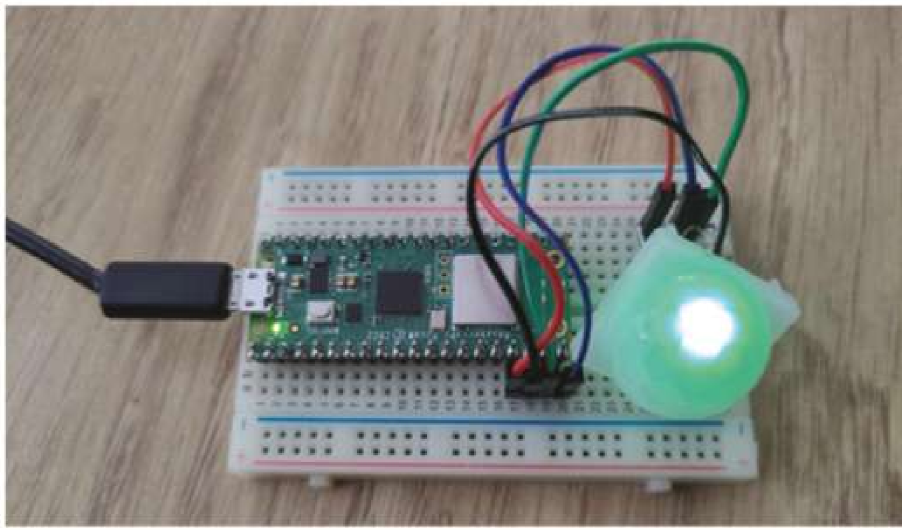
Go to **anvil.works** and sign up for a free account. We want to use slider controls for our lamp, which aren't available as a standard Anvil component, so we'll open up a demo app containing the Sliders library: go to **anvil.works/library/slider** and click 'Open in Anvil'.

The Sliders demo app only has two sliders and we want three, for red, green, and blue. From the right-hand Toolbox panel, drag a Slider custom component to just above the Reset button. Now select an existing 'Value:' element, press **CTRL/CMD + C** to copy it, and paste it under the new slider with **CTRL/CMD + V**. Copy and paste a '1' component to the right of it.

Click on each 'Value:' element in turn and change the text for it (in the right-hand Properties panel) to 'Red:', 'Green:', and 'Blue:' respectively. Set each '1' text element to '0'. Also, set the 'Blue:' element name to 'label_6, and the accompanying '0' element name to 'label_7'. For each slider, set the 'slider_max' value to 255.

08 Enable server uplink

Click the '+' button at the bottom left and select Uplink, then click Enable next to Server Uplink. This will generate a Server Uplink Key (ignore the client key) for the app which you



The cover from a PIR sensor makes a decent diffuser for the light from the RGB LED should paste into line 5 of your MicroPython code on Pico W so it can connect to it.

09 Adjust web code

We need to alter the web app code so the setting for each slider is sent to the red, green, or blue function in your Pico W program. Click on the Code tab and add the following lines (indented) to the `slider1_change` function:

```
    anvil.server.call_s("green",self.slider_1.
level)
    pass
```

Note that slider 1 is the middle one, which we've assigned to green. Similarly, for `slider2_change`, add the following two lines (indented):

```
    anvil.server.call_s("red",self.slider_2.
level)
    pass
```

Add a similar new function for slider 3:

```
  def slider_3_change(self, **event_args):
    self.label_7.text = self.slider_3.level
    anvil.server.call_s("blue",self.slider_3.
level)
    pass
```

Finally, add these lines to the `reset_btn_click` function in the web code:

```
    self.slider_3.level = 0
    self.label_7.text = 0
    anvil.server.call_s("red",0)
    anvil.server.call_s("green",0)
    anvil.server.call_s("blue",0)
    pass
```

10 Run both apps

We're now ready to roll. First, run your Pico W program in Thonny. You'll see messages in the Shell pane to show it connecting. Once it has, run your Anvil web app. You can now move the sliders to adjust the red, green, and blue components of your RGB LED to alter its shade.

To make it a little more effective, you may want to diffuse the LED's light. We simply placed the translucent plastic cover from a PIR sensor over it, which works fairly well. You now have a web-controlled mood lamp! Next time, we'll be reading Pico W sensor data in an Anvil web app.

mood_lamp_sliders.py

> Language: **MicroPython**

DOWNLOAD THE FULL CODE:

magpi.cc/github

```
001.  import anvil.pico
002.  import uasyncio as a
003.  from machine import Pin, PWM
004.
005.  UPLINK_KEY = "<put your Uplink key here>"
006.
007.  # We use the LED to indicate server calls and responses.
008.  led = Pin("LED", Pin.OUT, value=1)
009.
010.  # Set LED levels to zero
011.  redlevel = 0
012.  greenlevel = 0
013.  bluelevel = 0
014.
015.  # Set PWM pins for R, G, and B LEDs
016.  pwm13 = machine.PWM(machine.Pin(13))
017.  pwm14 = machine.PWM(machine.Pin(14))
018.  pwm15 = machine.PWM(machine.Pin(15))
019.  pwm13.freq(1000)
020.  pwm14.freq(1000)
021.  pwm15.freq(1000)
022.
023.  # Red LED control
024.  @anvil.pico.callable_async
025.  async def red(slider):
026.      redlevel = slider * 256
027.      pwm13.duty_u16(redlevel)
028.
029.  # Green LED control
030.  @anvil.pico.callable_async
031.  async def green(slider):
032.      bluelevel = slider * 256
033.      pwm14.duty_u16(bluelevel)
034.
035.  # Blue LED control
036.  @anvil.pico.callable_async
037.  async def blue(slider):
038.      greenlevel = slider * 256
039.      pwm15.duty_u16(greenlevel)
040.
041.  # Connect the Anvil Uplink. In MicroPython, this call
      will block forever.
042.  anvil.pico.connect(UPLINK_KEY)
```

Part 02

Raspberry Pi Radio: Add a DJ and jingles

On air in 3, 2, 1… You're listening to Raspberry Radio, the only station that plays your own music, all day and all night

MAKER

Sean McManus

Author of *Mission Python*, *Scratch Programming in Easy Steps*, and *Raspberry Pi For Dummies* (with Mike Cook). Get free chapters at Sean's website.

sean.co.uk

With Raspberry Radio, you're guaranteed to hear your favourite songs. The program creates a virtual DJ, who plays your MP3s at random, but introduces each one with some information about it. Every eight songs, there's a break for the news and weather. For an extra touch, you can add a Display-O-Tron HAT to show the artist and track name while it plays, like a DAB radio does. This project shows you how to make your Raspberry Pi speak, how to access the metadata in an MP3 file, and how to play music from Python.

01 Prepare your files

Raspberry Radio makes use of the **rr_newsreader.py** program from Part 1 in this series (see *The MagPi* issue #121, **magpi.cc/121**). Put **rr_newsreader.py** in the same folder as the **raspberry_radio.py** program from this issue. That folder should also have two subfolders: one called **music**, and another called **jingles**. You can download all the code for this project at **magpi.cc/raspradio**.

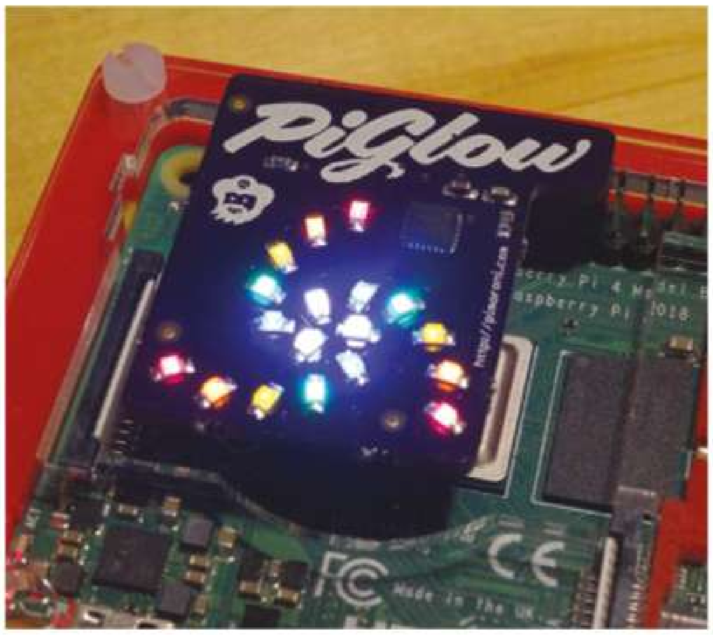
02 Install Python modules

For text-to-speech, we're using pyttsx3. Playsound will play our MP3s. Meanwhile, tinytag will read the metadata of the music files. By pulling out the artist name, song title, album, and year, we can get the virtual DJ to say something smart about each track before it plays. You also need to install the modules that the **rr_newsreader.py** program requires, if you didn't do that last issue. Open a Terminal window and enter the following commands at the prompt to download and install the modules, along with text-to-speech and media player software:

```
 pip install pyttsx3 tinytag playsound
requests feedparser
 sudo apt install espeak mopidy
```

03 Gather music and jingles

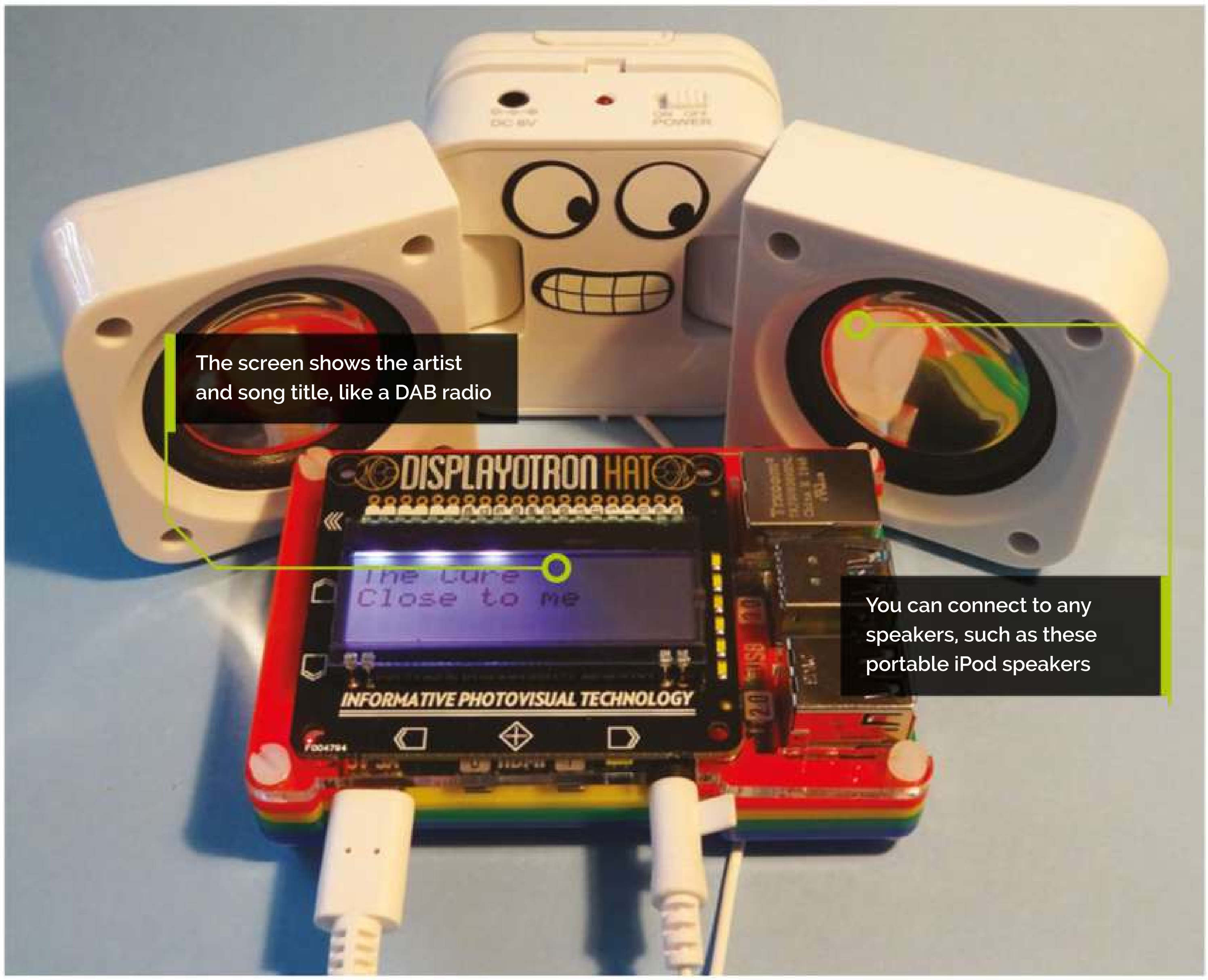
Raspberry Radio uses your own MP3s, which you should copy to the **music** subfolder. If you don't have an MP3 collection, lots of artists make their music available for free download on Bandcamp (find the author's at **magpi.cc/artificial**). To make it feel like real radio, we will top up the **jingles** folder. We recommend Music Radio Creative (**magpi.cc/freejingles**), which has lots of themed collections for free download. It's OK if your **music** and **jingles** folders have subfolders. Find a news jingle and store it beside the **raspberry_radio.py** program, not in the **jingles** folder. Call it **news_jingle.mp3**. We'll play it when the news is read out.



Using Pygame (instead of Playsound) and a PiGlow (instead of the Display-O-Tron), you can flash the lights while music plays. See the code at **magpi.cc/discolights**

You'll Need

- Some MP3 music files
- Some MP3 jingles
- Internet connection
- Display-O-Tron HAT (optional) **magpi.cc/displayotron**
- PiGlow (optional) **magpi.cc/piglow**



The screen shows the artist and song title, like a DAB radio

You can connect to any speakers, such as these portable iPod speakers

04 Edit the jingles

Radio and podcast jingles often have a slow fade out on them, because the (human) DJ speaks over the end of them. You can edit out silence or cut an excessive fade using Audacity. Install it from the Terminal using `sudo apt install audacity`. You'll find it in the Sound & Video category of your desktop menu. Open a jingle, and click the Play button in the top left to see where the fade becomes inaudible. Click and drag from that point to the end of the sound, and then use **CTRL+X** to cut the selected audio. Select File > Export to save your trimmed MP3 file.

05 Make Raspberry Pi speak

The instructions that make the computer talk are spread out in the program, so here's a simple demo that brings them all together. Enter these instructions in the Python Shell:

```
import pyttsx3
voice = pyttsx3.init()
voice.say("hello")
voice.runAndWait()
```

The `voice.say()` function queues up speech, but it isn't spoken until the `voice.runAndWait()` function runs. You can optionally set the rate (or speed) of the speech. In line 62, we chose 170. It's a bit slower than natural speech, so it's easier to understand. Lower numbers are slower still, and higher numbers are faster. We packaged up the speech instructions in the `output()` function at the start of the program. It also prints the messages to the screen.

> “You can edit out silence or cut an excessive fade”

06 Choose your DJ

There are several English accents you can choose from, including en-scottish, english-north, english_rp, english_wmids, english-us, en-westindies. As you can see, some use hyphens and some use underscores in their names. RP is short for received pronunciation and is the accent you hear on old BBC news reels. You can change your DJ's voice by adding an instruction like this:

Top Tip

Be selective

Indexing the files takes some time, so choose your favourite tracks and albums, rather than pointing the program at your entire iTunes library.

raspberry_radio.py

DOWNLOAD THE FULL CODE:

magpi.cc/raspradio

> Language: **Python 3**

```python
# Raspberry Radio from The MagPi by Sean McManus
import rr_newsreader, random, pyttsx3, os, sys
from playsound import playsound
from tinytag import TinyTag
import dot3k.lcd as lcd # Remove if not using Display-O-Tron

def output(text):
    print(text)
    voice.say(text)
    voice.runAndWait()

def broadcast_news_and_weather():
    playsound('news_jingle.mp3')
    date = rr_newsreader.get_date()
    output(date)
    news_headlines = rr_newsreader.get_news()
    for line in news_headlines:
        output(line)
    weather_report, temperature = rr_newsreader.get_weather()
    output(weather_report)

def index_directory(path, songs, perform_checks):
    print("Processing directory:", path)
    for entry in os.listdir(path):
        path_plus_entry = os.path.join(path, entry)
        if os.path.isdir(path_plus_entry):
            index_directory(path_plus_entry, songs, perform_checks)
        elif entry.endswith('.mp3'):
            tag = TinyTag.get(path_plus_entry)
            if perform_checks == False or \
                  (tag.title is not None and \
                   tag.genre not in ["Books & Spoken", "Christmas"] and \
                   tag.duration < 6000 and \
                   "live" not in tag.album and \
                   "live" not in tag.title):
                songs.append(path_plus_entry)
                print("Track added:", tag.title, "by", tag.artist, "from", tag.album)
    return songs

def play_songs(number_of_songs):
    for _ in range(number_of_songs):
        if random.random() > 0.4:
            jingle_to_play = random.choice(jingles)
            playsound(jingle_to_play)
        song_to_play = random.choice(songs)
        tag = TinyTag.get(song_to_play)
        dj_says = random.choice(
            [   f"What were you doing in {tag.year}? Here's what {tag.artist} was up to.",
                f"Here's a {tag.year} track from the album {tag.album}.",
                f"Fancy some {tag.genre} music? Here's {tag.artist}."
            ])
        output(dj_says)
        DAB_display = (tag.artist + ' ' * 16)[:16] \
                      + tag.title[:32]
        lcd.clear()
        lcd.write(DAB_display)
        playsound(song_to_play)

songs = index_directory("music", [], True)  # folder for music
jingles = index_directory("jingles", [], False) # folder for jingles
voice = pyttsx3.init()
voice.setProperty('rate', 170)
while True:
    broadcast_news_and_weather()
    play_songs(8)
```

```python
voice.setProperty('voice', 'english_wmids')
```

If you want to change the voice, we suggest you add the instruction near the end of the program, after you set its speed. You can change the voice back by setting it to `'default'`.

07 Playing MP3s

There are several different ways you can play music tracks from Python on Raspberry Pi. We're using the playsound module because it's easy to use, and the code is concise. You can play an MP3 using just two lines of code:

```python
from playsound import playsound
playsound('filename.mp3')
```

By default, playsound pauses the program until the sound has finished. That stops our DJ's speech and the audio files clashing. You can pass an additional value of False to start the sound playing without pausing the program.

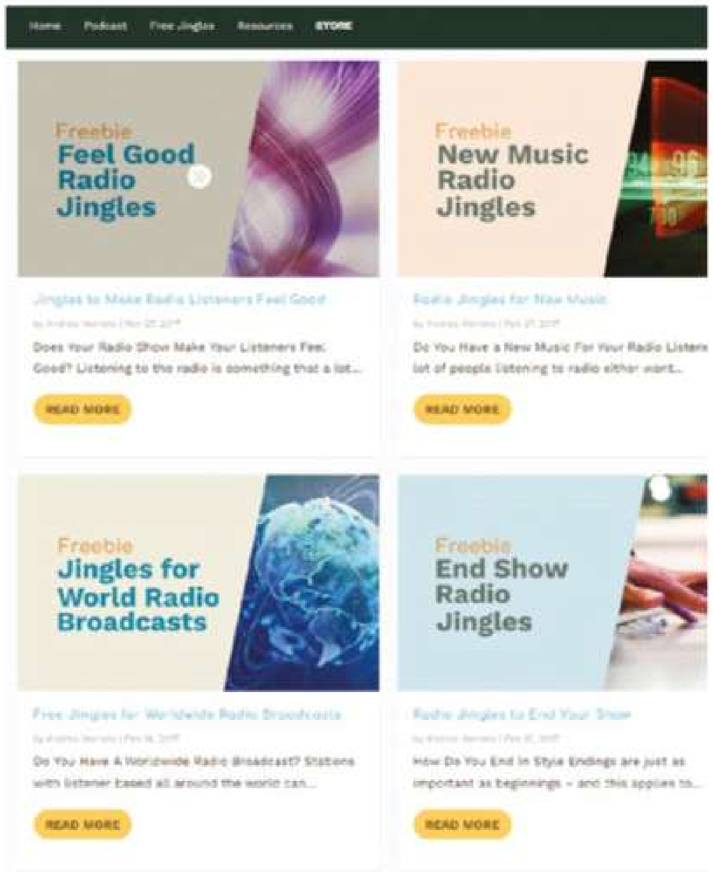
08 Indexing the music files

The `index_directory()` function creates a list containing all the music files in a directory and its subfolders. It might look familiar: we used a similar function to index images for ArtEvolver in issue 119. This time, we pass an additional True or False value to the function to say whether we want to perform quality checks. The quality checks ensure that songs are added that will work well on the radio. First, it checks they have a song title. Then, it excludes tracks that are in the Christmas or Books & Spoken genres, songs longer than 6000 seconds, and songs with 'live' in the album or track name. It's distracting when bursts of applause break through, although we do sacrifice some songs along the way (such as

Top Tip

Speed up indexing

You can speed up the indexing by deleting or commenting out the `print` statements in the `index_directory()` function.



Jingles to Make Radio Listeners Feel Good

Does Your Radio Show Make Your Listeners Feel Good? Listening to the radio is something that a lot...

Radio Jingles for New Music

Do You Have a New Music For Your Radio Listeners lot of people listening to radio either want...

Free Jingles for Worldwide Radio Broadcasts

Do You Have A Worldwide Radio Broadcast? Stations with listener based all around the world can...

Radio Jingles to End Your Show

How Do You End In Style Endings are just as important as beginnings – and this applies to...

Music Radio Creative provides lots of free jingle packs, made with professional voiceover artists and high-energy sound effects

Live to Tell by Madonna). When indexing the music folder, we want to make sure the songs follow those rules. The jingles don't need metadata (and probably don't have it), so we turn off the quality checks for them.

09 Extracting the music metadata

The tinytag module is used to extract metadata from the MP3 file in line 46. We can discover the song title, artist, album, year, genre, and duration. We've assumed that if the song title is present, then other metadata will be too. You can also find the composer, which is well-supported for classical music, but less so for pop and rock. The more accurate and the more complete your metadata is, the more authentic the DJ will sound.

10 Creating the DJ banter

The `dj_says` variable contains a randomly chosen phrase for the DJ to say before the song plays. We use f-strings to insert one or more tags into the phrase. Only a few examples are included here. The more you add, the less repetitive your DJ will sound. Have fun with it: it's easy to make robotic announcements like "This is Prince. Here's *Purple Rain*." It sounds more like a real DJ to say something like "What were you doing in 1984? Here's what Prince was up to!"

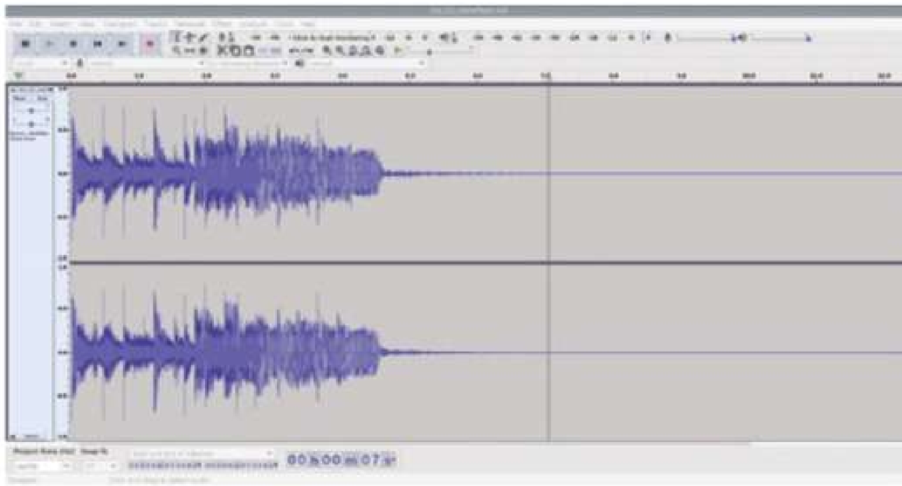
11 Making the DAB display

The best feature of our DAB radio is that it shows us the track and artist that's playing, so if you miss the DJ's introduction, we can still find out what it is. We've used a Display-O-Tron HAT to show the artist name and song name. If you don't have one, you can delete lines 5, and 53 to 56. The display is 16 characters wide and has three rows. The code takes the artist name, adds 16 spaces, and then keeps only the first 16 characters using `[:16]`. That ensures the artist name fills the first line and doesn't spill over. The song name is cut to its first 32 characters so it doesn't wrap from the bottom line to the top.

> "There's lots you can do to extend this project"

12 Build on it!

There's lots you can do to extend this project. You could remove songs from the list when they're played, so they don't get played twice. If you use Pygame to play the music instead of the playsound module, you can display animations while the music plays. The downside is that Pygame doesn't support MP3 files, so you'll need to convert your files to the OGG format. You can find some example code to cycle through the lights on a PiGlow add-on board at **magpi.cc/discolights**



This twelve-second jingle only has between six and seven seconds of audible sound. Using Audacity, you can trim it

MAKE | BUILD | HACK | CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
NEXT GEN 3D PRINTING
Why your next printer should be a Voron
Relays
Vinyl cutting
Aquaponics
KEYBOARD LEGO SHADOWS BRICKS

Build your own Poltergust G-00

Construct a 3D-printed, ghostbusting vacuum from Luigi's Mansion 3 with a Raspberry Pi Pico to make it extra cool

MAKER

Rob Zwetsloot

If Rob had a nickel for every time he's made a Luigi's Mansion Poltergust, he'd have two nickels, which isn't a lot, but it's weird that it's happened twice.

magpi.cc

You'll Need

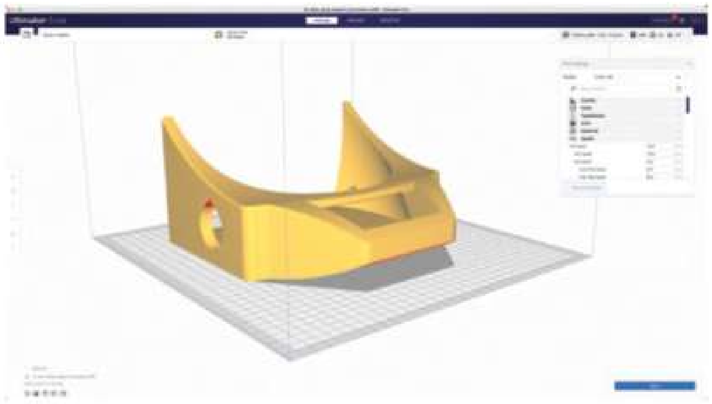
- 3D printed parts
- Small motor
- NeoPixel ring **magpi.cc/neopixel16**
- NeoPixel strips
- Mobile battery
- Wires
- Dremel
- Spray paint
- Leather strap
- Side release buckles
- Corrugated pond hose
- Various nuts and bolts
- Glue gun
- Soldering iron

Looking for a cool costume accessory this Halloween to really make your costume stand out? Pair this 3D-printed vacuum cleaner with an over-the-counter Green Plumber costume and you have a thematically correct costume that won't break the bank.

The Poltergust G-00 is the tool of choice of Luigi in Luigi's Mansion 3, and we thought we'd jazz it up with a Raspberry Pi Pico with lights and motors and more. It's moustache time.

01 Print your vacuum cleaner

In the past, Luigi's Mansion Poltergusts were easily made with old vacuums off of eBay. Just get one and you're good to go! However, the developers cheated a little bit with the Poltergust G-00 and, in this case, it is much easier and accurate to 3D-print one. The version available here by Cryzzgrantham is really well put together and free: **magpi.cc/poltergust1**. However, the handle and nozzle are quite small. If you'd like a bigger model, we used the one from Double A Cosplay on Etsy here: **magpi.cc/poltergust2**. It uses fewer parts, so it's not as easily customisable without further editing or power tools.



▲ The parts are ready to print but, if you have experience with 3D printing, it doesn't hurt to play around

02 Clean up your vacuum

The tolerances on these prints can be quite tight, and 3D printing isn't always perfect, so test out the part fits after printing. We took a rotary tool to the tyres, fan housing, and hose connector to make sure they all fit fine. The main vacuum body can slot together very nicely and mostly hold by friction, but be prepared to add hot glue later.

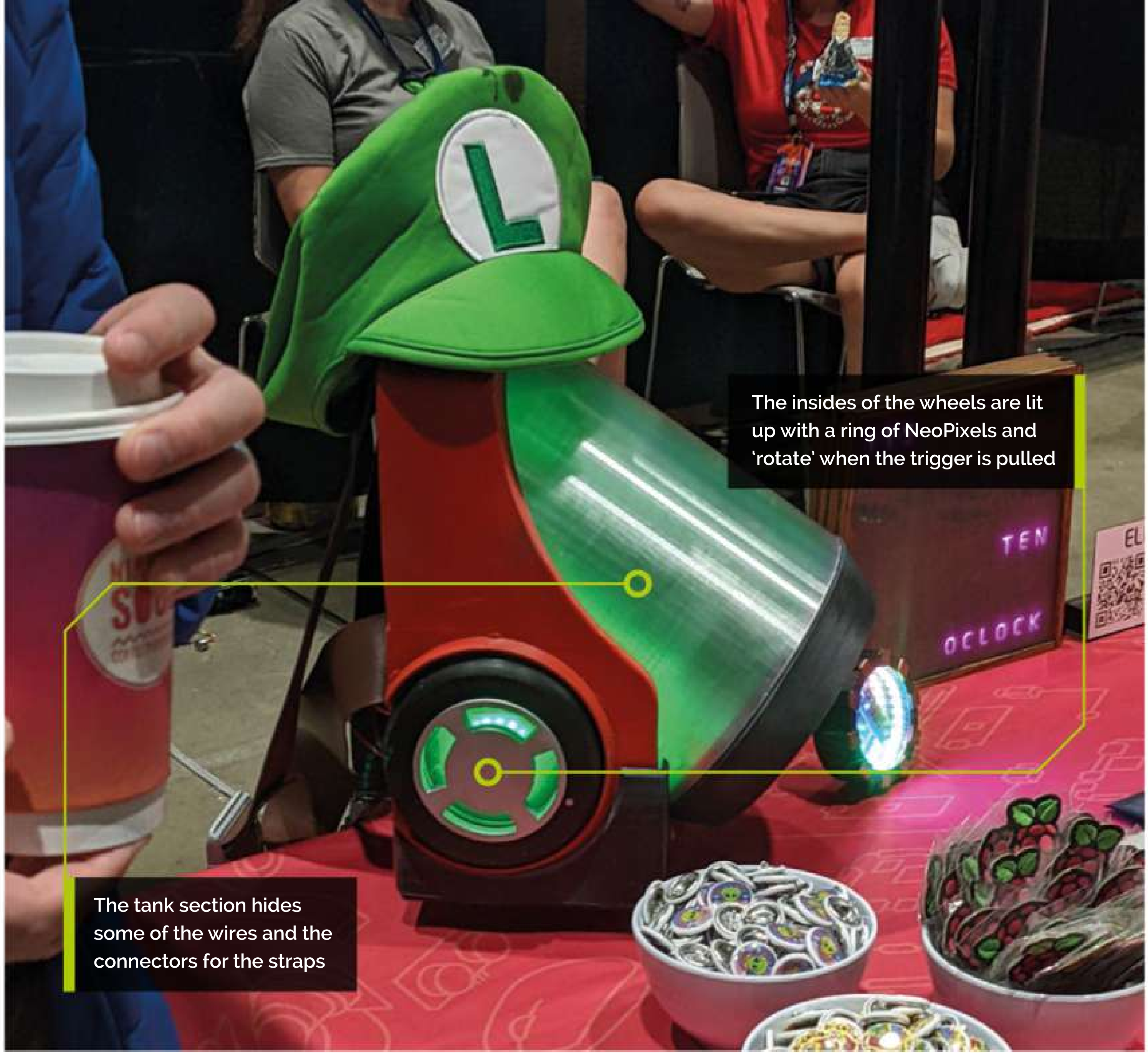
We made the fan housing larger than needed, as we plan to have the fan spinning when a trigger is pulled on the nozzle section.

As for the nozzle section trigger – if you use the smaller handle, the trigger on that can be more easily rigged to be depressed. On the larger one, we drilled a couple of holes into the trigger housing so we could add a button, with wires going through the hose section.

03 Smooth your vacuum

Depending on your level of time and care for detail, you can now spend a couple of days getting the prints to look very smooth. There are many ways to do this; however, the tried and true method of applying a thin layer of gesso, sanding it down, and then repeating until the surface is smooth is great as it also holds paint very well.

You can use various materials to fill the gaps on the main vacuum case as well, such as using a 3D print pen to add similar material, or some kind of Polyfilla works well. Do this before you start your gesso layers, though.



The insides of the wheels are lit up with a ring of NeoPixels and 'rotate' when the trigger is pulled

The tank section hides some of the wires and the connectors for the straps

Warning!
Power Tools

Make sure you are careful when use rotary tools, drills, or power sanders when modifying 3D prints

04 Paint your vacuum

Prime, paint, topcoat. For our vacuum, we went with a grey primer so that some of the edge details would stand out more. A couple of coats of that all over and allowed to dry will allow the paint to adhere better. Luckily, most of the bits are separated by colour, and you just need red, black, and silver for them. Two or three coats of each colour is a good idea. For the alternate handle, we painted it all silver first, let that dry, and then used masking tape to cover the bits we wanted to remain silver before going over it again with black. Use a craft knife to trim the tape to get sharper edges on the paint.

After that's all done, a clear or gloss topcoat will finish it off. We suggest gloss to match the game. Remember to do all of this outside or in a well-ventilated room.

> "You can use various materials to fill the gaps on the main hoover case"

05 Build a test circuit

It's always a good idea to test your circuit before installing it into any project. If you follow along to **Figure 1**, it shows you how to wire up your circuit to a Pico on a breadboard. All the LEDs can be powered off of the 5 V pin on Pico, as it gets power directly from whatever powers Pico. However, a resistor is very useful to make sure they don't draw too much power. Eventually you'll need to install these components, and Pico, into the vacuum housing at the bottom. You can either solder them onto Pico, or just glue the breadboard into there for easy maintenance.

06 Installing the code

With the Pico circuit set up, you can start testing the code. Download the code from **magpi.cc/poltergust** and then head to **magpi.cc/circuitpython**. Here, we're going to

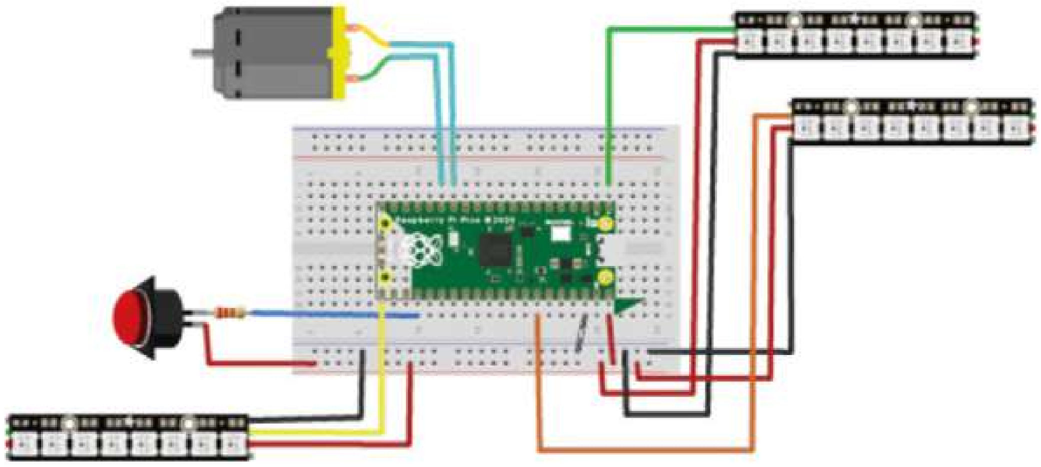
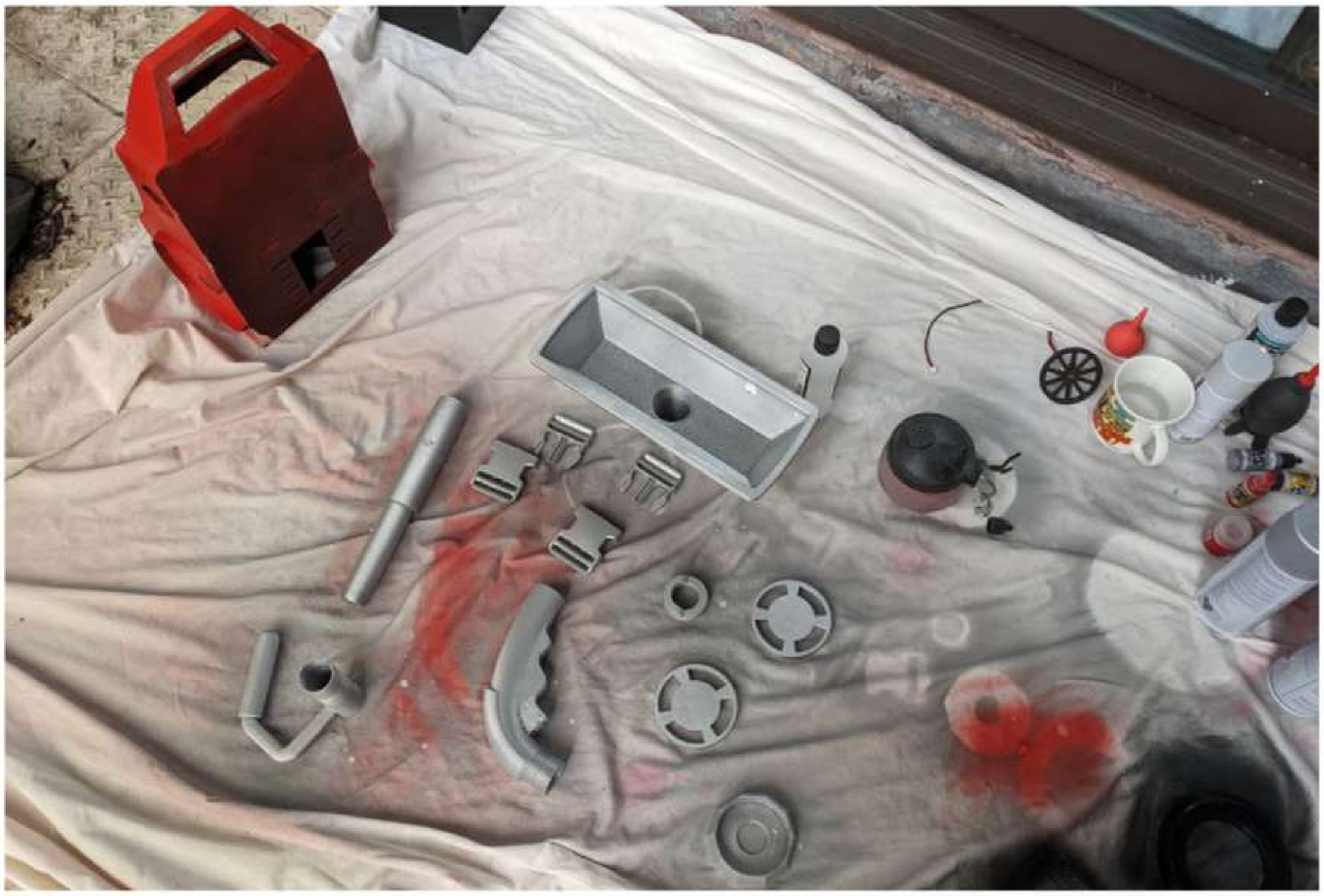


Figure 1 The components are arranged in sections to make separation easier. Bottom left is the handle, bottom right is the tank, top right is the wheels, and then the motor is on its own

Top Tip

3D printing

We used an 8% infill on the print, and clear PLA for the tank as well. While not very smooth, it does give a nice eerie glow.



We used a mixture of airbrush and canned spray paints to colour the parts

Top Tip

Expanding foam

If parts of your print aren't great, you can easily fix it with an old cosplay trick: expanding foam. Apply it, let it dry, carve it with a knife then, after about 24 hours, you can sand it down.

download the files needed to allow Pico to run using CircuitPython, which has some fancy LED animations we're going to use. You'll need to then hold the BOOTSEL button down on Pico before plugging it into your PC and then dropping the CircuitPython UF2 file onto the drive that pops up. It will disappear and then come back as CIRCUITPY, where you can then add the **code.py** file from the code bundle you downloaded.

Finally, put the .mpy MicroPython library files into the folder **lib** on Pico.

07 Customise your code

The Poltergust code has two states: idle and active. By this we mean it will behave a certain way when turned on and nothing is happening (idle), and with a button you'll be able to change the behaviour of LEDs and a motor (active).

Always check the fit of 3D printed parts

The LEDs on the handle and wheels are linked so only require one pin, and the tank glows whatever is happening. The motor has been turned off in the code and you should be careful if using it. The throttle is set low so you can test it out yourself without making it go dangerously fast.

You can also change up the number of LEDs on the first line of each section.

08 Install your electronics

This specific print has a couple of hollow support rods that run through the body, and holes that access the wheels as well. These are perfect for running wires through. Run the handle LED wires down the side that fits to the hose connector, and the wires for the tank light on the other side. The tank is fairly enclosed, so use your judgement on where to glue those LEDs. Stick the NeoPixel circles to the inside of the wheel spokes and run the wires to the main empty compartment on the bottom. You just need to hook it all into Pico via soldering or with a breadboard.

09 Start construction

Two 50 cm leather straps, cut in half, have been attached via 5 mm nuts and bolts so they hold

DOWNLOAD THE FULL CODE:

magpi.cc/poltergust

A table full of Pico-powered projects includedour own Poltergust, glowing throughout the day

firm in the housing and turn as well. Slot the side clasp buckles through them on the other end. A lot of the rest of the vacuum cleaner comes together with friction. However, parts like the wheels, wheel rim, tank, etc. need to be hot-glued down. In our build, we removed the inner wheel and inner fan section so that we could more easily add the electronics. These are the circles with the pegs sticking out of the middle.

> The LEDs don't take up a huge amount of current, so a decent mobile battery will easily last you a long time

10 Finish construction

We've run the wire for the handle LEDs and button through the hose and now need to stick all of that to the handle, and then seal the hose to both ends. The bigger handle is made from four pieces, which all slot together quite nicely without gluing. However, if you don't need to take it apart again, you could always use glue.

11 Finish the straps

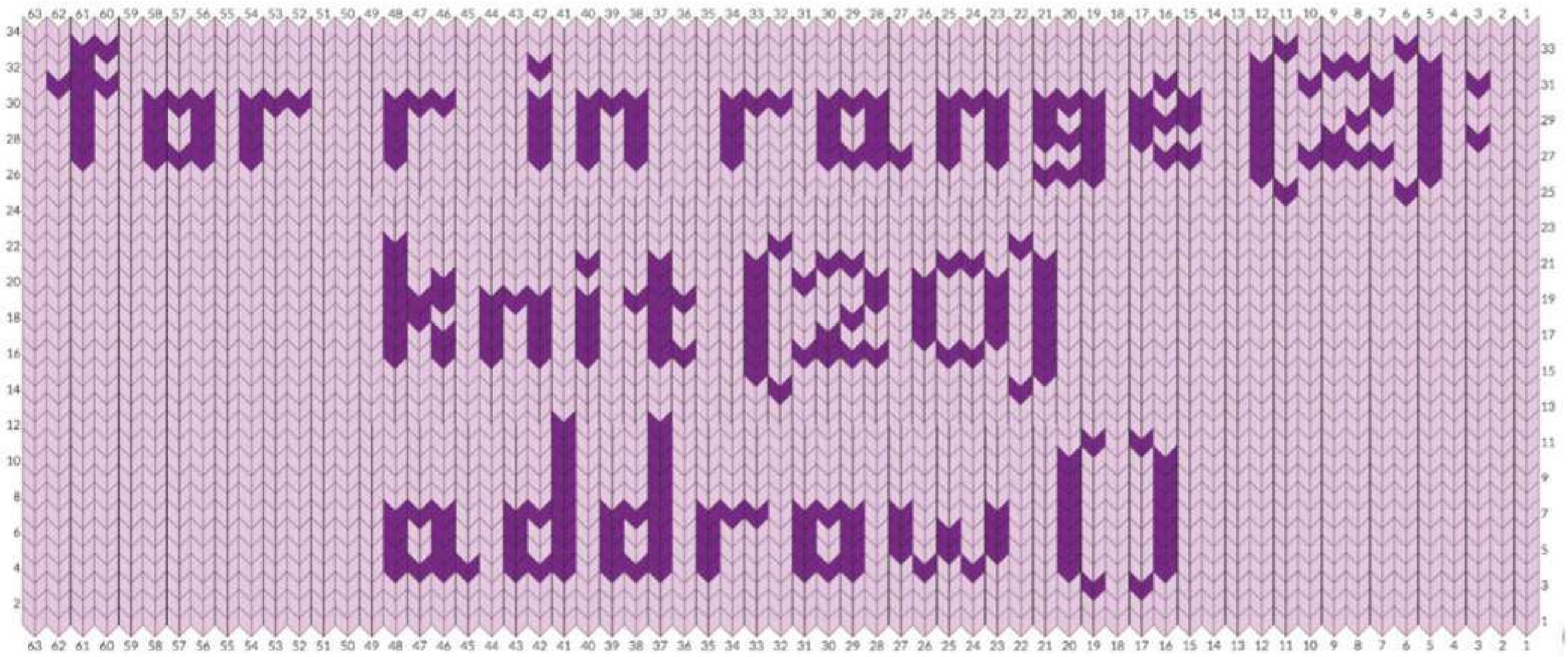
Put the vacuum on and attach the side clasps – you should be able to figure out how much of the leather you can snip off. We suggest doing it from the top buckle so that the buckles themselves sit high on your chest, like in the game.

Once that's done, you can sew the ends together on all the straps, or use glue – the glue may not hold super-well on leather or pleather, though.

Don't forget the Luigi costume to go with it!

12 Powering and testing

We had a USB cable dangling from the hole in the back of the case connected to a mobile battery in the back pocket of our dungarees. Plugging it into that should have the full system turn on and start glowing and spinning and such. Test out the button to make sure it works properly. The LEDs don't take up a huge amount of current, so a decent mobile battery will easily last you a long time – we went a whole day with barely any loss in power or light!



A stitch in code

The intriguing relationship between the fibre arts and coding

Nicola King

@holtonhandmade

Nicola King is a freelance writer and sub-editor. She firmly believes that purchasing arts and crafts goodies, and actually sitting down and making something, are two completely different pastimes.

Have you ever considered how a knitting or a crochet pattern resembles coding, and the many parallels that exist between the fibre arts (an umbrella term which includes knitting, crochet, spinning, weaving, and yarn dying, to name but a few) and coding? These two seemingly unrelated fields have an interesting history together and, in this article, we will take a look at the fascinating similarities that really illustrate how much they have in common – spreading, we hope, an appreciation of the fact that coding and the fibre arts have many more similarities than you may ever have previously considered.

A CLOSE-KNIT RELATIONSHIP

Let's begin with a little history, which illustrates how coding/programming and the fibre arts are entwined. To see one of the most famous examples of this, we can simply look at the Jacquard loom, patented in 1804 by French weaver Joseph-Marie Jacquard. The Jacquard system was way ahead of its time, and enabled unskilled workers to weave intricate patterns easily: the central principle being the use of a chain of interchangeable punched cards. In weaving, a single weft thread is passed over and under a set of warp threads. Before this invention, workers had to manually raise and lower wrap threads in order to create any kind of fabric pattern, and this was an extremely slow process.

The cleverly designed cards of the Jacquard loom were fed into the Jacquard mechanism at the top of the loom and had small holes punched in them, which held the instructions for the weaving pattern. They basically controlled which warp threads should be raised to allow the weft to pass underneath. So, this was a very early example of instructing a machine to carry out an automated process, using a binary system (with either a punched hole, or no punched hole), where the information could be read by the loom and reproduced as required. The use of the replaceable punched cards to control a sequence of operations is widely held as an important step in the history of computing, as Charles Babbage (1791–1871) was inspired by the Jacquard loom's use of punch cards to program his Analytical Engine of 1837, considered by many as the first modern computer design. So, we can begin to see a link

Above Fibre arts have a lot in common with coding, as we shall explain

GRAPHGHANS

There is a stitch in crochet called C2C or 'corner to corner', and each C2C stitch basically creates a small square or block. This means that C2C is a handy stitch to know if you want to design a pixel crochet blanket for example (also known as graphghans – afghans made from graphs), made up of a series of squares where each square represents one pixel of the image. You could write your own simple pattern for this on a piece of graph paper and then stitch it up. Or, you could use a website such as Stitch Fiddle (**hsmag.cc/PixelCrochet**) and create your chart there, where you can upload a picture to the site, or create a new design yourself. C2C is similar to how computers store images in grids of coloured squares.

What we are also illustrating here is that crocheters and knitters are very happy to use tech help, including websites and apps, in their designs if it makes the process easier – is your image of the average knitter or crocheter evolving at all? It's not just about the yarn these days…

here between the two, on first sight, seemingly very different worlds.

ILLUSTRATIONS OF CRAFTING CODE

Let's now look more closely at the parallels between coding and the fibre arts, and perhaps the easiest thing to do to illustrate these is to take a look at some very simple patterns which, when knitted and crocheted, will produce the swatches shown in **Figures 1** and **2**, respectively.

SIMPLE KNIT PATTERN

- CO 16 stitches
- R1 – P2, [K5, P2] to end of row
- R2 – K2, [P5, K2] to end of row
- R3 – P2, *K2tog, YO, K1,YO, Sl1, K1, PSSO, P2; rep from * to end of row
- R4 – Repeat R2
- Repeat R1 – R4 until work measures 10 cm in length
- Bind off

(Abbreviations: CO – Cast On, R – Row, K – Knit, P – Purl, K2tog – Knit two stitches together, YO – Yarn Over, Sl – Slip, PSSO – Pass slipped stitch over)

SIMPLE CROCHET PATTERN (UK CROCHET TERMINOLOGY)

- Ch 15
- R1 – [2 Tr, 1 Ch, 2 Tr] in 5th Ch from hook, *skip 3 Ch, [2 Tr,1 Ch, 2 Tr] in next Ch, repeat from * to last 2 Ch, skip 1 Ch, 1 Tr in last Ch, turn
- R2 – 3 Ch, skip first 3 Tr *[2 Tr, 1 Ch, 2 Tr] in 1 Ch sp, skip next 4 Tr, repeat from *, end by skipping last 2 Tr, 1 Tr in next Ch, turn
- Repeat R2 until desired length

(Abbreviations: Ch – Chain, Tr – Treble, Ch sp – Chain space, R – Row)

These are forms of code, knitting and crochet code: sets of instructions used to complete several

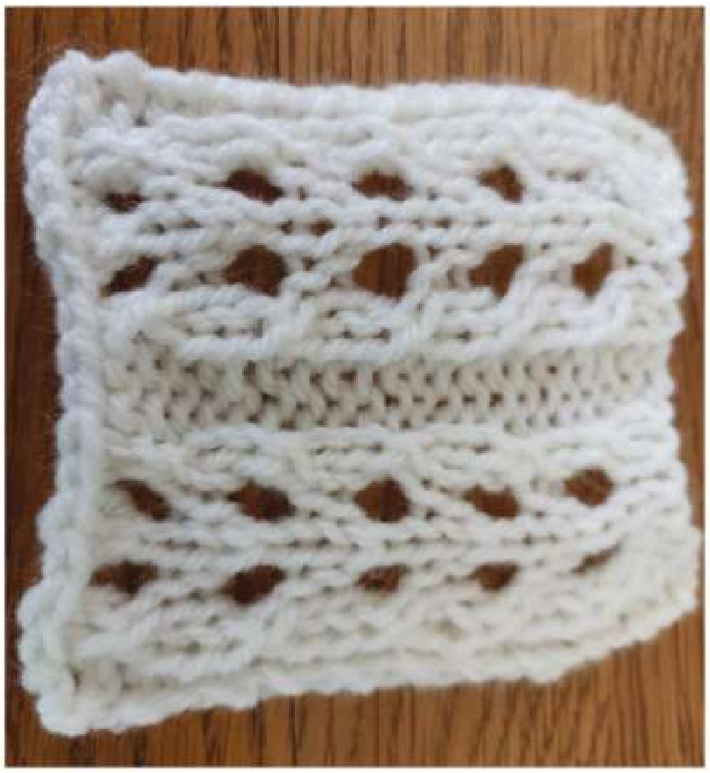


Figure 1 The knitted swatch illustrates a simple lace design achieved by following the dictated code and various commands therein. In knitting, stitches stay on the needle and are active for a whole row, and you use either single point, double point, or fixed circular needles

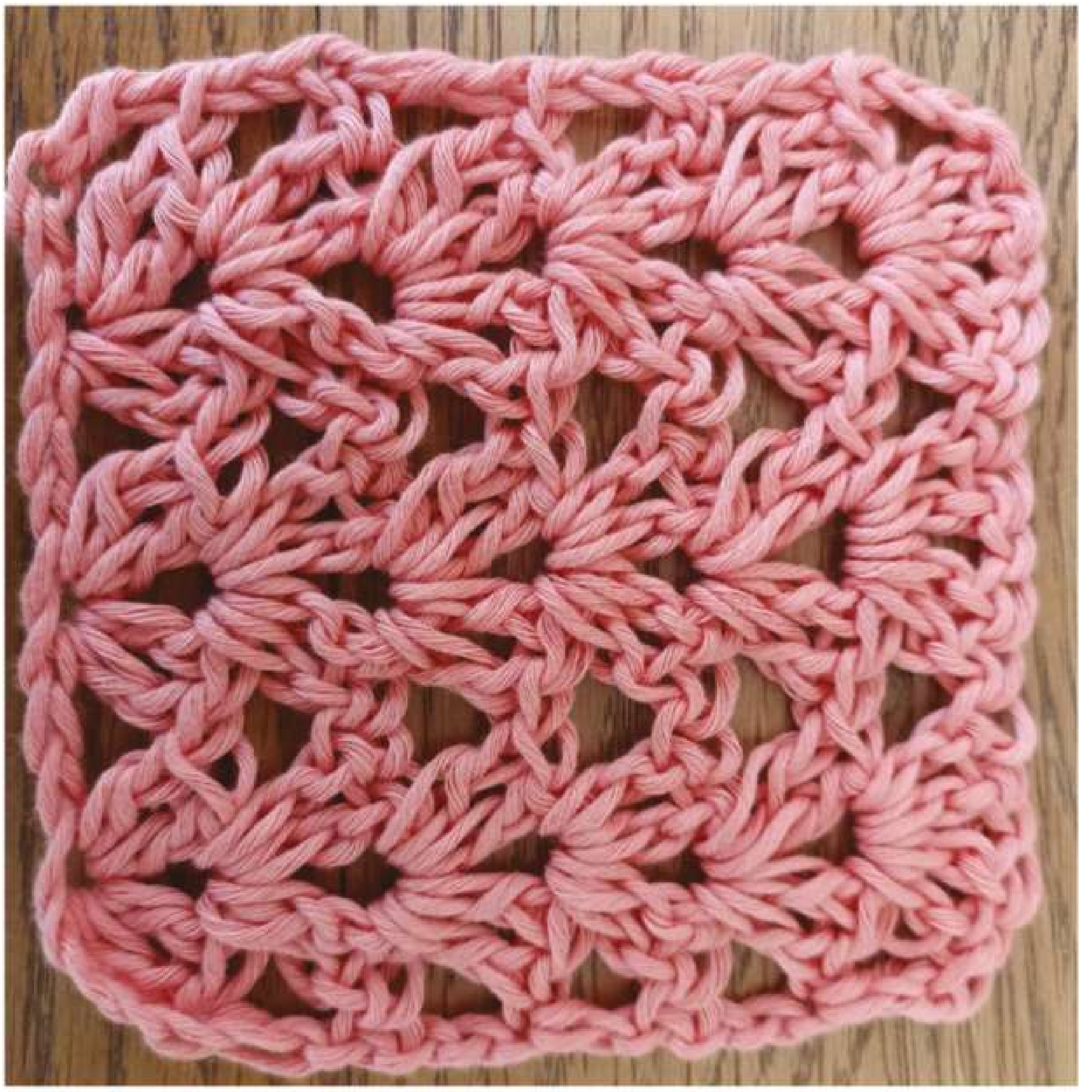


Figure 2 The crocheted swatch – a different effect is achieved from following the crochet code. In crochet, you complete each stitch one at a time, so only one stitch is live, and work with just a single hook

Above We downloaded the free (open-source) pattern (code) from the designer (developer), channelled the Force through our hook and yarn, and hey presto… The Child! (hsmag.cc/BabyYoda)

rows of knitting or crochet. Note particularly, in both patterns, the use of brackets and asterisks, which are also of course used in coding – here, these are indicating a group of stitches that will be repeated, or a specific instruction to be carried out in a certain stitch such as, in the crochet example, completing two trebles, one chain, and two trebles in a single stitch. Designing a pattern is basically just like writing a program. You can think of the knitter or crocheter as the computer, translating the code, executing functions in the code line by line, in the correct order. Knitters and crocheters won't generally refer to these instructions as code; in fact, it may not occur to many of them that it is code, but any knitted or crocheted item, such as a scarf, is created one stitch at a time from such a piece of code. OK, if you're not a knitter or crocheter, these patterns won't make a huge amount of sense, and the purpose of this article is not to teach you how to knit or crochet. Also, knitting and crochet use very different techniques and tools to reach their end-products, but you can see from the patterns on the previous page that code plays a very important part in these two crafts – makers need the correct, error-free code in order to perform the function.

CRAFTY CORRELATIONS

So, looking specifically at the handcrafts of knitting and crochet, it can be suggested that the numerous comparisons between coding and these two fibre-based techniques include the following:

1. **Standard notation**

Knitting and crochet patterns are written using a standard notation describing precisely the actions required to achieve the desired outcome – a series of detailed steps to create an object or a piece of fabric made from stitches. Similarly, computer programs are also written using a standard notation describing the steps a computer needs to take to solve a problem.

2. **Loop-the-loop**

As already mentioned, if you take a look at the patterns on the previous page, you will see the use of asterisks and brackets. In knitting or crochet, these basically indicate the beginning of a stitch sequence which often needs to be repeated until you reach a

certain point, such as the end of the row, for example. Sounds familiar… a little like a loop in coding perhaps?

3. Terminology

Both the spheres of coding and fibre arts use their own languages or vernacular – a unique syntax that seems to make little sense unless you understand/research the terms involved. A few baffling knitting expressions include rib, cables, brioche, intarsia, frogging, and entrelac, while crochet lingo includes seemingly unfathomable terms such as amigurumi, back post double crochet, filet, mosaic… and so on. But, as with anything, once you understand that jargon, it all falls into place.

4. Open source

In the world of the fibre arts, patterns are sometimes paid for, but can often be found for free. Fibre artists, like coders, are a generous bunch, and many people make their patterns available to all on sites such as Ravelry (see separate box). Hundreds of knitting and crocheting podcasters are also willing to share their knowledge for free on YouTube in videos showing how to crochet a Baby Yoda, how to weave a wall-hanging, or how to knit a pair of socks. There is a real sense of community in the fibre arts world, as the many forums and online communities will attest – much like the world of programming and coding.

> " Note particularly, in both patterns, the use of brackets **and asterisks, which are also of course used in coding** "

5. Maths, maths, and more maths

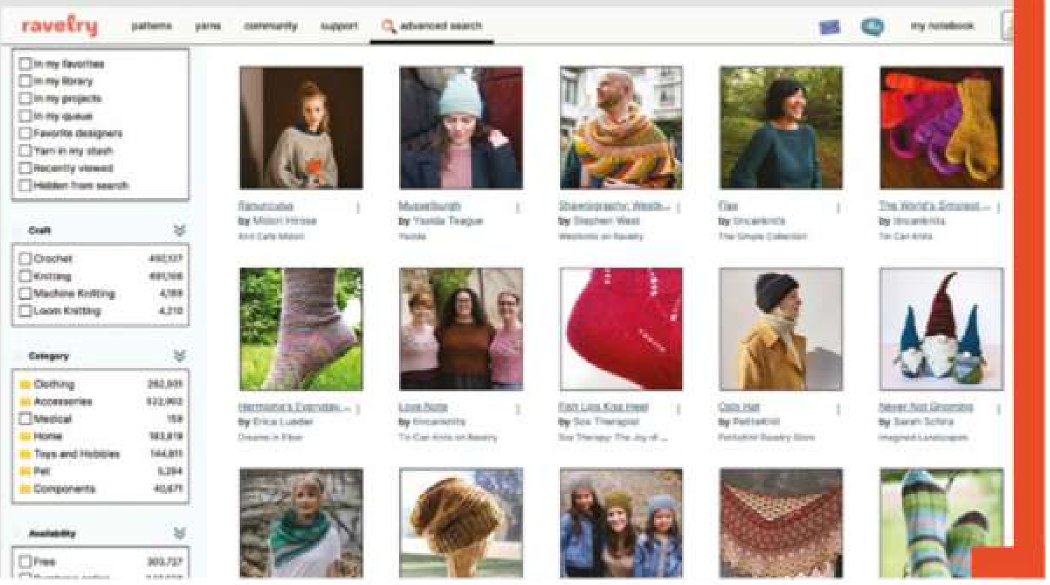
When this author is knitting and crocheting, at the same time, she is also counting. It's about order, accuracy, and precision. Following the code, she is processing that algorithm that a designer has created in order to instruct her on how to achieve the end goal. When executing the code, she needs to count (a lot!). Stitch counts are critical, so she needs to follow the designer's code precisely, or that crocheted version of Bernie Sanders at the inauguration will have the wrong-sized head, (yes, a talented individual had the ingenuity to design such a pattern – **hsmag.cc/Bernie**). 'Ssshh, I'm counting' is a line that this author's family are more than familiar with. Believe it or not, repetitive counting is also very calming as it's so meditative.

RAVELRY: A CODING AND KNITTING COLLABORATION

There are many stereotypes associated with knitting and crochet but, these days, the average age of a fan of these two crafts, in particular, is much lower than in years gone by. Banish (please!) any thoughts of little old ladies in rocking chairs – fibre artists are cool, on-trend, tech-savvy, and want to regularly communicate with fellow creators online.

Consequently, tools have developed to meet the inherent demand for a form of 'social networking' specifically for the yarn arts. Ravelry (**ravelry.com**) was created in 2007 by Jessica and Cassidy Forbes from Boston, Massachusetts, and now has over nine million (yes, you read that figure correctly!) users worldwide. Interestingly, one half of the founding partnership was originally a knitter, and the other was… a coder! Their complementary skills are still used today to keep this gargantuan network functioning at its optimum. Nearly half a million knit and crochet patterns, for example, can be downloaded from the site, some free, some paid for, thus generating an income for thousands of pattern designers and sustaining Ravelry so that it can continue to unite fibre artists.

Users can utilise the organisational tools, share patterns, stashes, thoughts, ideas, discussions, and projects. It's a fibre-loving community using a massive database of patterns and yarns – you can filter your searches on hundreds of criteria and get instant access to exactly what you're looking for. It's free to join, and a great place to visit if you've lost your crochet or knitting mojo, as you're guaranteed to want to start at least 150 different projects once you start looking for inspiration.



Above A vast database of thousands of patterns at your fingertips, or should that be needle tips?

This author recently downloaded a free crochet pattern in which the maths didn't add up, based on what the code was telling her to do and the required stitch counts at the end of each row. The pattern appeared to not have been tech-edited, so she emailed the company concerned and rewrote the bothersome incorrect elements of the pattern. She's determined to write more of her own crochet code now, as it's really not that hard!

6. Refactoring and checking for 'bugs'

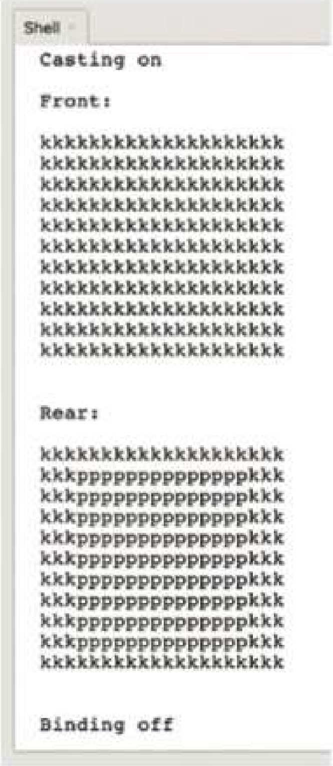
Just as coders will refactor code to try and improve the design of a piece of software, fibre artists do the same thing, as it's all in the finish. Most paid-for patterns, and many free patterns too, will be →

LEARN A NEW SKILL AND FURTHER READING

If you would like to experience the many therapeutic benefits that knitting, crochet, weaving, or any other fibrous form of creativity can offer (mindfulness, reduction of anxiety and stress, improved cognitive function, sense of control, sense of achievement… to name just a few), then there are many options:

- YouTube videos – there are hundreds to choose from, and it's how this author taught herself to crochet. Try: **youtube.com/c/BlossomCrochet/videos**, **youtube.com/c/RJKnits/videos**, or **youtube.com/c/SpruceandLinen/videos**
- Ask a friend/family member to teach you – nothing beats a hands-on lesson with an expert
- Join an online lesson. Try: **creativebug.com**, **skillshare.com**, or **udemy.com**
- Buy a beginner kit – these come with instructions and everything you need. Try: **weareknitters.co.uk**, or **bytomdaley.com/collections**
- Get yourself a book from the library – we suggest you grab a 'How to'-type publication along with a stitch dictionary, if you are learning crochet or knitting

When you get that 'light bulb' moment, and everything finally clicks into place, you'll feel such a sense of achievement and you'll be away (oh, and start making space for some yarn, fibre, tools… it's a rabbit hole). You might even start designing your own patterns…also known as coding!



```
Casting on

Front:

kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkk


Rear:

kkkkkkkkkkkkkkkkkkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkppppppppppppppkkk
kkkkkkkkkkkkkkkkkkkk


Binding off
```

Figure 3
The Python program prints out a chart of stitches for front and rear

rigorously tested by test knitters or crocheters, and by tech editors, to make sure that there are no errors in the pattern. Much like coders and developers, designers will search for bugs in the system – the end goal in a knit/crochet piece of coding is documentation that delivers what it promises, and only once it has been debugged can it be released, or deployed in software terminology. As in coding, pattern designers will use trial and error to achieve their desired outcome. Designers will rewrite their code so that they explain the pattern, in as few lines as possible, very similar to the DRY (Don't Repeat Yourself) principle of software development. Fibre arts patterns and coding are both trying to communicate complex instructions as simply as possible.

7. Creativity is inherent

Fibre artists are, by definition, creative. They often tweak patterns, add their own touches, choose appropriate yarn colours, and basically express themselves through their yarny creations. Take, for example, yarn bombing (**hsmag.cc/YarnBombing**) – a creative, edgy, expressive form of street art, spreading colourful fibre joy. This author has recently found a pattern for crocheting toppers for UK postboxes, and hopes to create one soon for her local postbox. And, yes, coders are also creative – they need to possess a technical creativity in order to write any piece of software. It can be suggested that coders are aiming for a form of beauty in their code, bug- and error-free, just as fibre artists want to see a beautiful end-product too.

8. Tech skills are also essential

It's a given that coders require technical expertise, but fibre artists have those skills too. Being able to read a pattern and translate it is a technical skill, as is knowing how to knit or crochet in the first place, knowing what yarns to use for each pattern, measuring gauge correctly so that the new jumper you are constructing fits you properly, or modifying patterns to include your own personal slant, for example. (See 'Graphghans' box.)

> **You can do both – you can code and program and you** can knit or crochet or spin or weave

9. Continuous learning curve

As in all areas of life, both coders and fibre artists are constantly facing challenges to develop, to learn more, and to hone their skills, as there are always new elements to master. Fibre artists will be learning new stitches, techniques, about new yarns, how to read charts, how to fix errors, while coders need to keep up with new languages, frameworks, libraries, and so on.

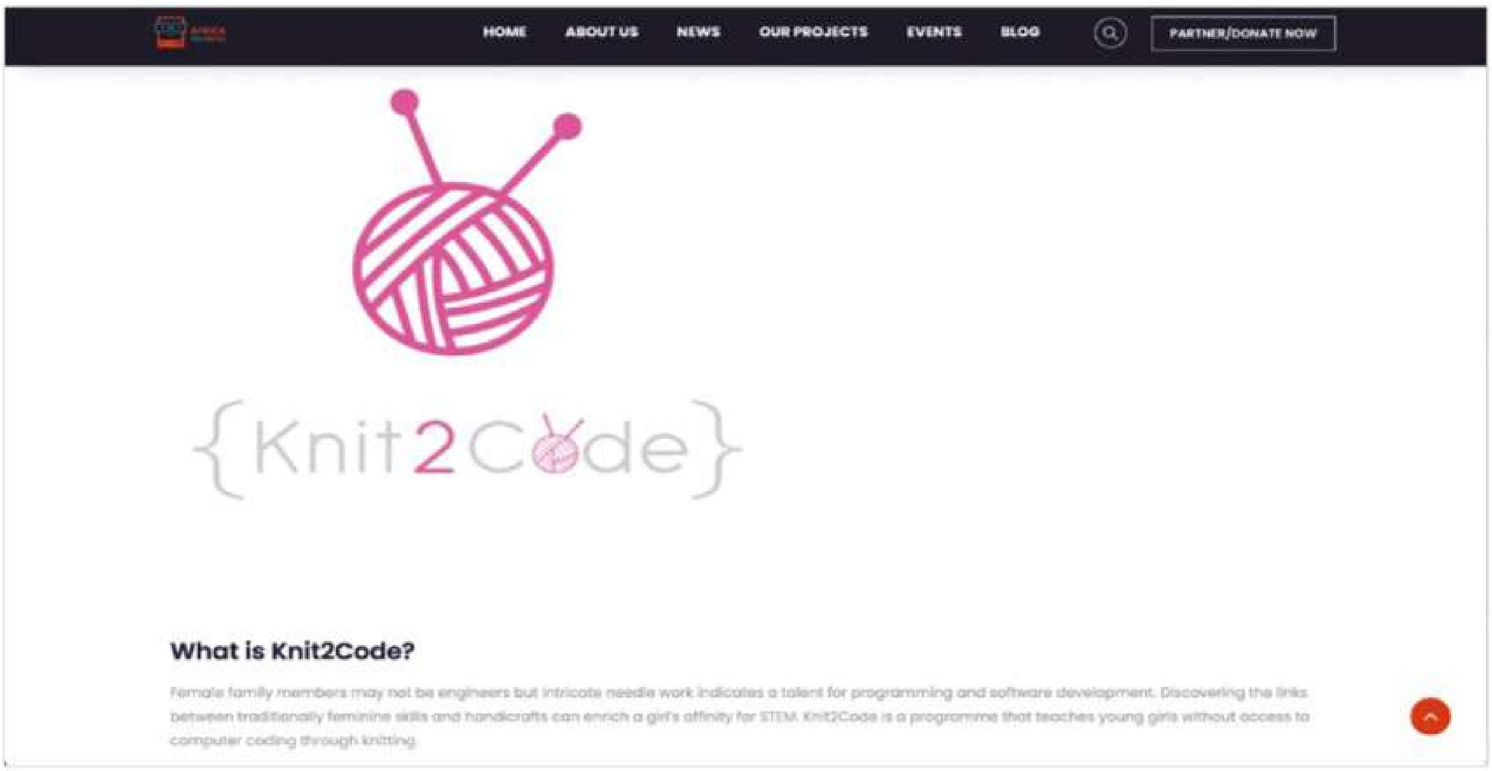
Perhaps a key point to make here is that you can do both – you can code and program and you can knit or crochet or spin or weave. You can be both creative and technical at the same time. Coding and the fibre arts are not mutually exclusive pastimes – would you be surprised to find a coder or developer who likes to knit, or a professional crochet designer who also coded for fun? Based on the many similarities discussed here, we wouldn't be surprised at all… and there are indeed many people who happily do both.

AN EXPERIMENT

So, just to see what would happen, this author decided to take a very basic knitting pattern and turn it into Python code (with the help of her code-loving husband!)

The original pattern she drafted was as follows:

- CO 20 stitches
- R1 & R2 – K
- R3 – K
- R4 – K3, P14, K3



What is Knit2Code?

Female family members may not be engineers but intricate needle work indicates a talent for programming and software development. Discovering the links between traditionally feminine skills and handicrafts can enrich a girl's affinity for STEM. Knit2Code is a programme that teaches young girls without access to computer coding through knitting.

Left Africa Teen Geeks' Knit2Code programme brings together girls of different generations to learn, relearn, or enjoy knitting and, as a by-product, the basics of computing

- Repeat R3 & R4 until work measures 10 cm in length
- Final 2 rows – K
- Bind off

You can find the link to the resulting Python code here (**hsmag.cc/issue57**). As you can see, there's quite a lot of repetition/looping here, along with a conditional ('until work measures 10 cm in length'). In our Python code, we created functions for knit and purl, which simply add a letter 'k' or 'p' to a 'row' string the number of times specified in a 'stitches' parameter. We also created a special 'row4' function which calls the knit and purl functions to execute its mixture of knitted and purled stitches.

An 'addrow' function adds each completed row to a 'front' or 'rear' string – since in knitting, the side of the work is reversed with each row, we alternate this.

The first 'for' loop knits two rows (akin to 'R1 & R2 – K' in the pattern). A 'while' loop then repeats rows 3 (knit 20) and 4 until the 'length' variable (increased with each row) is 10 or greater. We finish with another 'for' loop to knit two more rows. At the end, we print the 'front' and 'rear' strings, which results in a chart of stitches for the front and rear of the work (**Figure 3**).

To conclude, if you'd like to read more, we really suggest that you take a look at this article from The New York Times (**hsmag.cc/YarnProgrammable**) where the ideas of physicist Dr Elisabetta Matsumoto, that 'yarn is a programmable material' and that 'knitting is coding', are explored. It's mind-blowing stuff, looking specifically at the mathematics of knitting. Here, we have really only just skimmed the surface of the coding and fibre arts entanglement, but hope we have piqued your interest in this fascinating area. □

COVERT CODE

Did you know that you can encode secret messages into that hat or cardigan that you're knitting? Well, this is something that was used to great effect during wartime – yes, unassuming knitters became useful spies, using steganography (a way of hiding a message physically in plain sight) and passing on valuable information in their pattern of stitches to the people who could use the information effectively in the war effort. It was relatively straightforward to encipher information into a fabric using, for example, Morse code. When you think about it, it makes complete sense – knitting consists of two kinds of stitches, knit (flat stitches resembling a 'v') and purl (horizontal bumps) – and it is the chosen combination of these that makes a pattern, and is relatable to binary code.

To illustrate, during World War I, Belgian Intelligence agents sought the help of knitting women who could see train stations and tracks from their windows. They asked them to monitor the passing of German Imperial trains – in their knitting, these inconspicuous-looking women would drop stitches or add extra purl stitches and so on, as a form of predetermined code, to signify how many artillery trains had passed, if a train full of troops had gone through, etc. They then passed their completed fabric to the Belgian Resistance, who could decode the fabric, based on their pre-agreed stitch meanings. Had they been discovered by the enemy, these knitters' lives would, of course, have been in peril. It must have been a successful form of information gathering, as coded knitting played a part in WWII as well. This is a truly fascinating subject area if you are interested in history and knitting espionage, and you can read more here: **hsmag.cc/KnittingSpies**.

Above Who'd have thought that a humble ball of yarn and two wooden sticks could communicate vital intelligence?

Build a Mini Magic Mirror

Shrink down a Raspberry Pi smart mirror to fit in smaller spaces

The magic mirror is one of those great Raspberry Pi projects that looks cool, isn't too hard to make, and is actually a bit practical too. The software is all basically taken care of, yet is very customisable for those who have the know-how to go further with their projects.

Smart mirrors are too useful to just have them limited to one big mirror in your home, though, so we've collected some smaller versions you can try out to put above your sink, or even on your desk. Seeing is believing.

Magic Mirror basics

How to build any kind of smart mirror with Raspberry Pi

01 Build a frame

Before anything else, you'll need an old/spare TV or monitor for your magic mirror. You'll need to build a frame to the size of the display, so it's no use starting without one. With the ability to take measurements, you can now get some wood from your local hardware store, or get a matching frame from IKEA or the internet to fit it in. You'll also need some two-way mirror acrylic sheet or film to cover the display to make it truly reflective.

02 Program

The software for a lot of magic mirror projects uses the MagicMirror code from **magicmirror.builders**. It's easy to install and customise and, from a fresh Raspberry Pi OS install on Raspberry Pi or Raspberry Pi Zero, you can go to the Terminal to download the software with:

```
  git clone https://github.com/MichMich/
MagicMirror
```

Head to the MagicMirror docs to see the full installation instructions and customisation options: **magpi.cc/mirrorstart**.

03 Hang it up

You'll need to find a place with suitable access to power to hang up your mirror and make it work. Use screws rather than picture pins wherever possible as the entire setup will be quite heavy, and it's best for the project and anything near it to not fall off the wall.

You can customise it with extra modules for different functions, connect it to your calendar, and much more for the true future living experience.

Tuesday, December 13, 2016
15:10
22:29
-0.6°
You look

Mini Mirror Projects

Make your smart mirror smaller

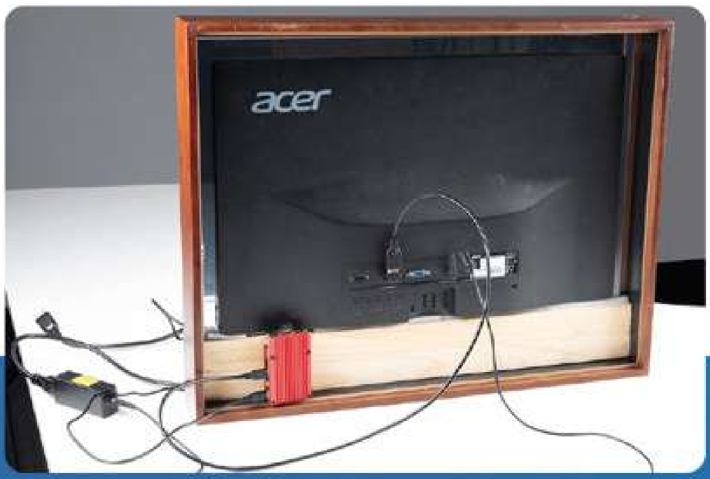
Magic Mirror with Raspberry Pi

MAKER: **SparkFun** | **magpi.cc/smallmirror**

This version of the magic mirror uses a smaller computer monitor and a deep shadow box frame to create a landscape-orientated mirror that can easily fit on a desk or side table. It was inspired by the busy personal life of the maker, and how they and their partner could use some kind of ‘battle station’ to keep track of their plans.

The shadow box it uses is not quite the same size as the monitor though, and so, some space has been filled in with wood so that it looks a little more natural. This is something you’ll have to contend with on any magic mirror project that uses premade frames but, with reflective film added on, it’s a lot less noticeable than it would be otherwise.

Finally, this tutorial has some great tips on customising your mirror’s layout and modules, tailoring them to your personal requirements.



The wood at the bottom helps fill in the space and raise the monitor up, and right-angle connectors help keep the whole system flat to the wall

Smart Mirror using a Raspberry Pi Zero

MAKER: **Brad Morton for LinuxScrew** | **magpi.cc/smallmirrorzero**

While you can definitely make something small with a standard Raspberry Pi, using a Raspberry Pi Zero really allows you to bring the size of a smart mirror down. This one, also landscape in orientation, uses a very small 5-inch screen, all dropped into a frame from IKEA. The display is mounted to the back of a thin piece of mirrored acrylic that fills up the entire space within the frame.

While you can basically hook up a Raspberry Pi Zero and stop there, this project shows you how to add a button that will sleep and wake the screen – saving on power and also letting you use the magic mirror as a full mirror. It requires a little extra hacking and the addition of a Python script, but it's not all that difficult to do.

▼ With the right wood finish, this mini mirror can easily fit in with any decor

Info screen projects

Want the data but not the mirror? Here are some alternatives

MagicMirror Alarm Clock

While not really a mirror, this alarm clock does use the MagicMirror software to make a very informative alarm clock with easy customisation.

magpi.cc/magicalarm

Digital Information Screen

Using software called Yodeck, maker Syed Khairi was able to create a useful info screen for his science department.

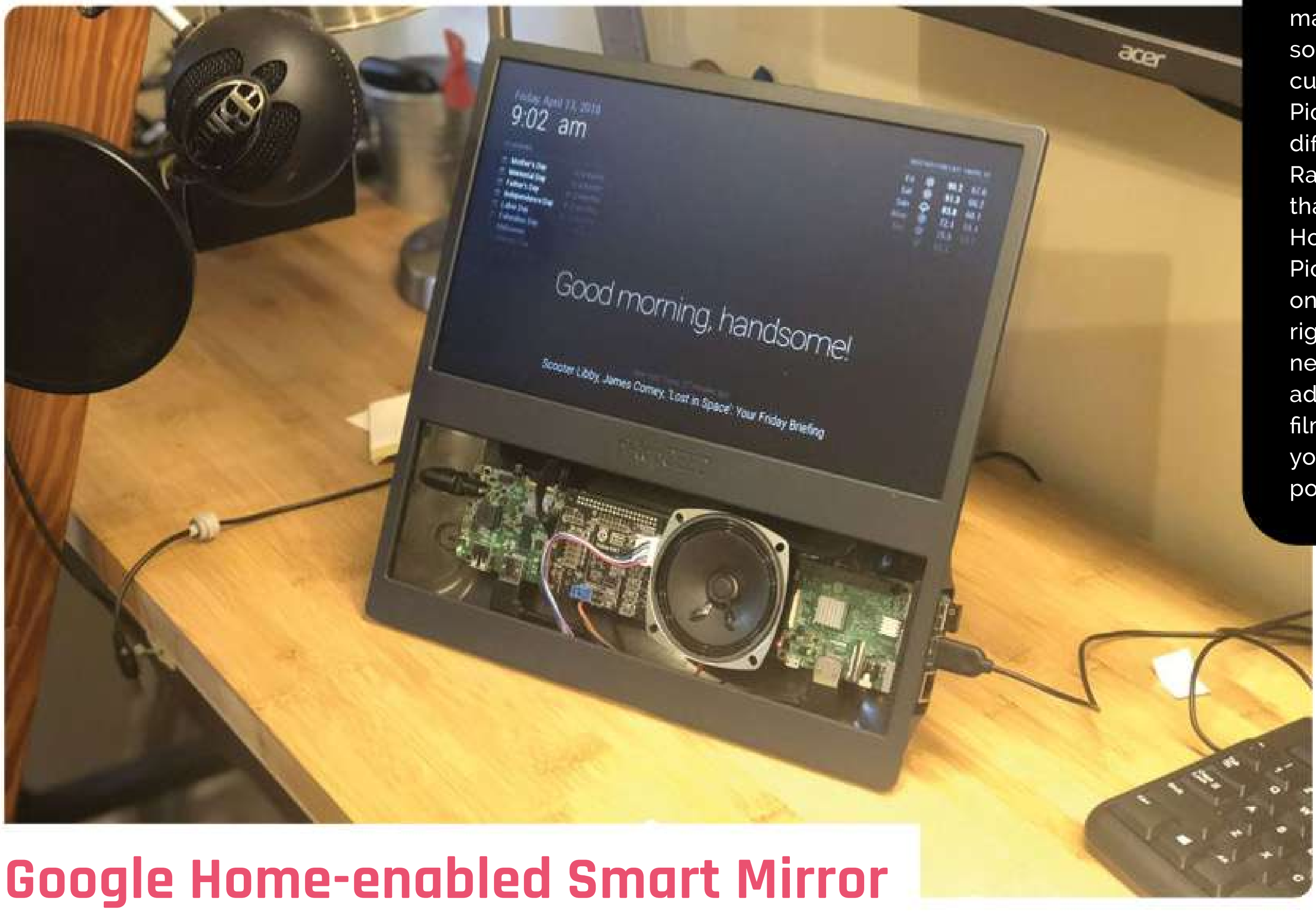
magpi.cc/discreen

13:16:23
Mo, 21 Jan 20
MAKING ENGINEERING HOT 2018
Have a good week
Y11 & 13 EASTER MOCK EXAM COUNTDOWN
37d

Chromium Kiosk

Kiosk mode in Chromium allows you to create a selection of panels and functions that limit the user to certain functions – such as looking at their calendar.

magpi.cc/chormiumkiosk


9:02 am
Good morning, handsome!
Scooter Libby, James Comey, 'Lost in Space': Your Friday Briefing

Raspberry Pi Pico displays

Unfortunately, the main MagicMirror software does not currently work with Pico – it's quite different from Raspberry Pi, so that makes sense. However, you can get Pico to display info on a screen with the right code – all you need to do is then add two-way mirror film or acrylic and you'll have a Pico-powered smart mirror.

Google Home-enabled Smart Mirror

MAKER: **Zach** | **magpi.cc/voicemirror**

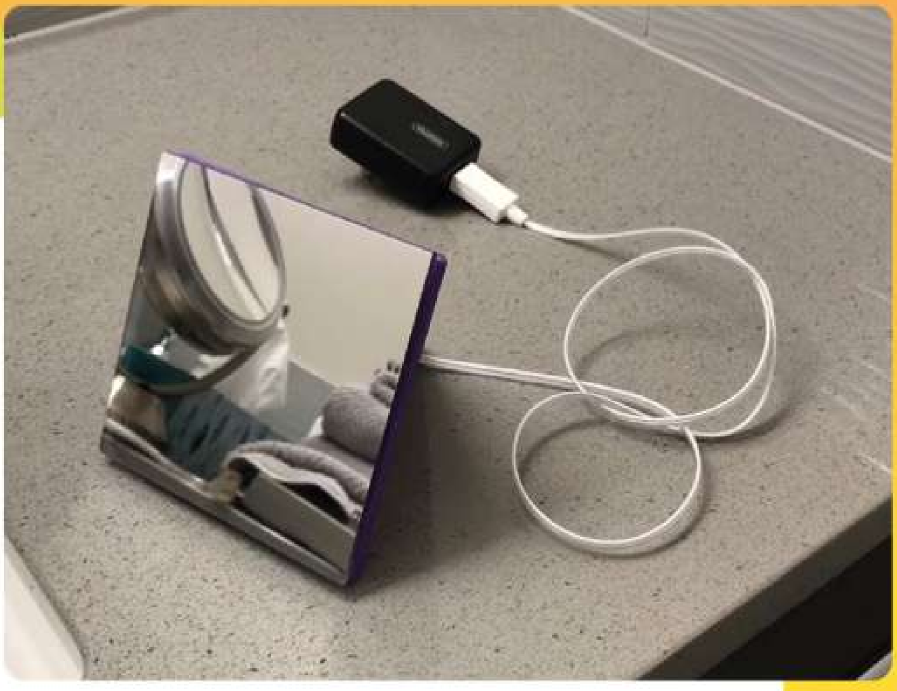
Adding voice commands to a magic mirror is very cool, especially when it uses tech that was once given away by *The MagPi*! The AIY Voice Kit is an easy way to add Google Home-like voice control to a Raspberry Pi project, and we quite like this one that also repurposes an older pi-topCEED. It allows you to efficiently use space, as it's already designed to be quite slim yet still hold a Raspberry Pi inside.

It can also be done in an afternoon! Zach reckons it took him about two hours, and we can see why – all he needed to do was add the Voice Kit, the MagicMirror software, and the two-way mirror – which is cleverly attached with magnets to make it removable for easy maintenance. There's no soldering either – everything just snaps in.

The trickiest part is getting the AIY Voice Kit to work with the software, but Zach has a great troubleshooting guide for you if you start having issues.

Due to the shape of pi-top, the final mirror is square (13˝ by 13˝), but it's quite unique-looking

We like the idea of taking this into the bathroom to shave with



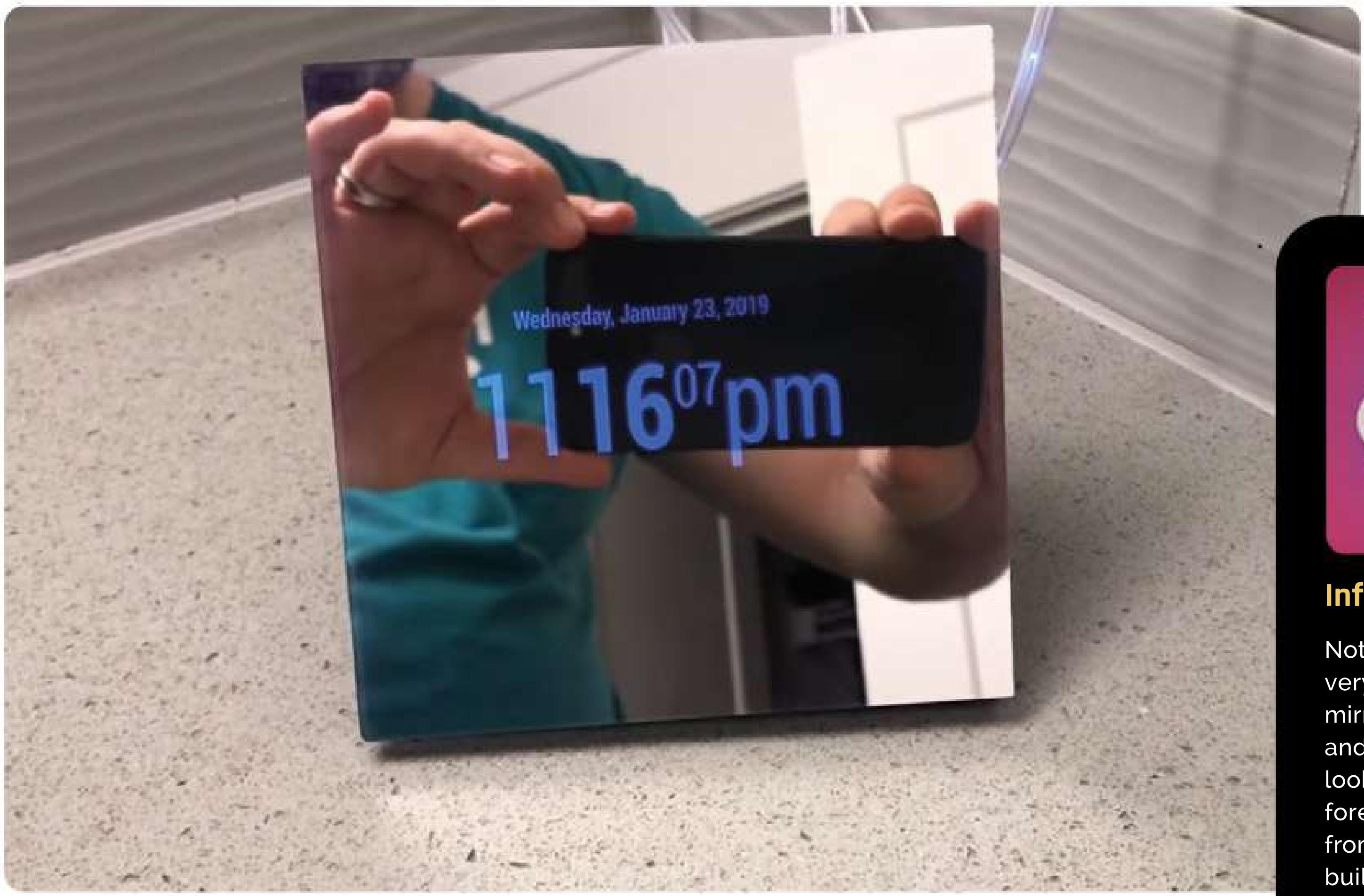
Mini Magic Mirror

MAKER: **GeekToolkit** | **magpi.cc/minimirror**

This magic mirror project truly is mini – it uses a 3.5˝ screen, and a 115 mm square two-way mirror. Don't worry about the mixing of inches and metric, especially when this whole project will cost you from $60/£50 to make!

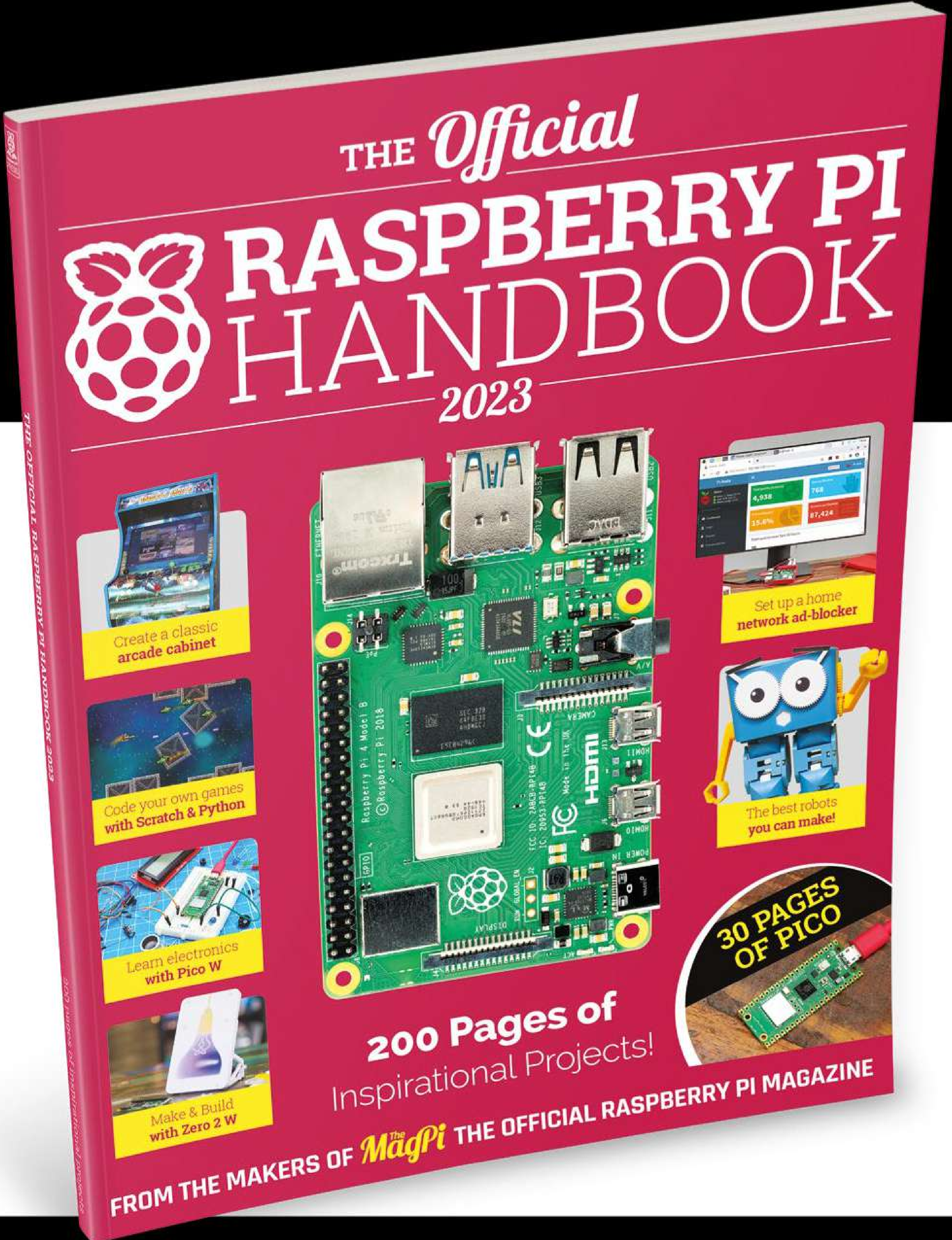
The pyramid-style stand for the screen is great for multiple reasons. Firstly, it's very stable; while the whole setup isn't very heavy, it still needs to be sturdy. Secondly, it is hollow, allowing for more than enough room for a Raspberry Pi Zero to sit inside.

The stand is made up of two 3D-printed parts that neatly connect together, and also allows space for you to plug the mirror into the mains – or to a mobile battery if you want to take it around your home. It's a very cool, very low-effort project that has many uses.



Infinity mirror

Not very smart but very cool, infinity mirrors use mirrors and lights to make it look like they go on forever. Toby Roberts, from Raspberry Pi, built this and others with a Raspberry Pi Pico inside and powering all the lights.



THE *Official*

RASPBERRY PI HANDBOOK

2023

Create a classic **arcade cabinet**

Code your own games **with Scratch & Python**

Learn electronics **with Pico W**

Make & Build **with Zero 2 W**

Set up a home **network ad-blocker**

The best robots **you can make!**

30 PAGES OF PICO

200 Pages of Inspirational Projects!

FROM THE MAKERS OF *The MagPi* THE OFFICIAL RASPBERRY PI MAGAZINE

THE OFFICIAL RASPBERRY PI HANDBOOK 2023

DeMoor Orrery
INTRODUCING
PICO W

Tufty 2040

▲ The rear features the battery connector, edge connector, and power and boot /USR buttons

▶ Pimoroni ▶ **magpi.cc/tufty2040** ▶ From £24 / $24

With a colour LCD screen, is this the ultimate interactive name badge? **By Phil King**

SPECS

DISPLAY: 2.4 in colour IPS LCD display, 320×240 pixels

POWER: JST-PH battery connector (input range 3–5.5 V), USB-C

FEATURES: 5 × user buttons, LED, Qwiic/STEMMA QT port, breakout edge connector (I2C, UART, SWD), 8MB flash storage

The Tufty 2040 badge is based on Raspberry Pi's RP2040 microcontroller chip, as used on Pico. It goes one better than the Badger 2040 (reviewed in issue 116, **magpi.cc/116**) with the use of a 2.4-inch colour LCD screen in place of a monochrome e-ink display.

With a rapid refresh rate, the LCD enables the Tufty to do a lot more, including showing animated text and graphics. A simple game is even included as one of the preloaded examples, selectable via a menu when you turn it on. Others include wavy scrolling text, Pride and retro badge layouts, and an old-school 'Sketchy-Sketch' drawing tool. As on the Badger, control is via five programmable user buttons.

Connecting the Tufty 2040 to a computer via USB enables you to program it in MicroPython or C++. The PicoGraphics library makes it relatively easy to write text and draw shapes. JPEGs can also be rendered, and sprites imported from a sprite sheet.

Wear it well

For portable use, a JST-PH battery connector accepts input from 3 V to 5.5 V. An optional Accessory Kit includes a 3×AAA battery pack, Velcro pad to fix it to the rear, and lanyard. A less bulky alternative is to use a LiPo battery, although the Tufty has no circuitry to charge it. Naturally, power drain is higher than using an e-ink display: around 100 mA in total. The on-board light sensor could be used to auto-dim the display via PWM, however.

A neat bonus feature is the Qwiic/STEMMA QT port which can be used to connect I2C sensors and other add-ons. So you could even use the Tufty 2040 as a data display instead of a badge.

> With a rapid refresh rate, the LCD enables the Tufty to do a lot more

▲ The Tufty 2040 displaying the Pride badge example on its colour LCD

Verdict

The crisp colour screen makes for a super-cool interactive name badge that is versatile and fairly easy to program.

9/10

Automation 2040 W

SPECS

POWER:
From 6 V to 40 V via screw terminals

INPUTS/ OUTPUTS:
4 × digital inputs, 3 × 12-bit analogue inputs, 3 × sourcing outputs, 3 × relays

FEATURES:
2 × user buttons, reset button, status LEDs, 2 × Qwiic/STEMMA QT ports, unpopulated header

▶ Pimoroni ▶ magpi.cc/auto2040w ▶ £59 / $58

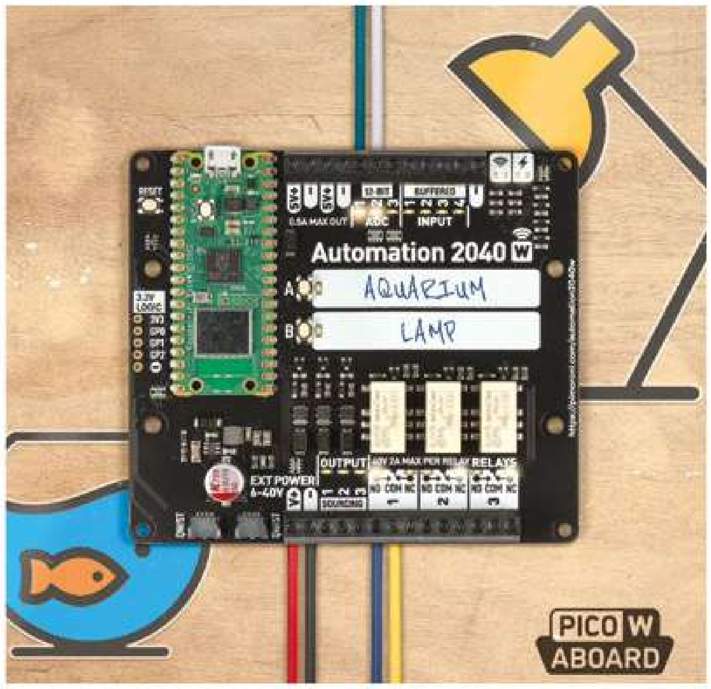
With a Pico W aboard, this is a home automation powerhouse. By **Phil King**

Unlike other home automation boards we've seen, the Automation 2040 W is a standalone affair with a pre-soldered Pico W aboard. So, all you need to supply is the power: from 6 V to 40 V, via two screw terminals.

Functionality-wise, it's similar to Pimoroni's Automation HAT, although that could only handle up to 24 V. Tolerant up to 40 V, the Automation 2040 W is compatible with 12 V, 24 V, and 36 V systems for all your home automation needs.

At the bottom of the board, three relays (all with the option of a normally closed or open connection) can output up to 40 V at 1 A, or 24 V at 2 A. Next to them are three (switchable) sourcing outputs that can provide up to 4 A at up to 40 V for things like larger motors, solenoids, and pumps.

Along the top are three 12-bit analogue inputs and four buffered digital inputs, all tolerant up to 40 V. There's also a pair of 5 V outputs for lower-voltage components. Finally, there's a couple of Qwiic/STEMMA QT ports for breakout sensors, and an unpopulated header for a few of Pico W's pins.



The Automation 2040 W has a wide range of 40 V tolerant inputs and outputs

You can program the on-board Pico W in MicroPython or C/C++ using Pimoroni's libraries

Let there be light

One really nice touch is the inclusion of a status LED (wired directly) for every input and output, so you can see when it's active – ideal for debugging programs (in MicroPython or C/C++), even when external hardware isn't yet connected. We also like the two programmable push-buttons with white space next to them to write labels.

> One really nice touch is the inclusion of a status LED for every input and output

Best of all is Pico W's own wireless connectivity, which means that you can easily set up a web interface (an example is included in Pimoroni's Getting Started Guide: **magpi.cc/auto2040wguide**) to control and monitor your devices.

Verdict

A little expensive, but then it is a standalone device and will work with almost all home automation systems.

8/10

10 Amazing: Cases

Keep your Raspberry Pi, or Raspberry Pi Pico, safe with these great enclosures

You may have noticed that a Raspberry Pi is sold essentially naked. However, you can easily get a case for your Raspberry Pi and give it some extra protection from the elements and spilled drinks. Here are ten of our favourites, including some nice cases for your Pico.

▲ SmartiPi Touch Pro

Case and stand

This enclosure allows you to create an entire Raspberry Pi system and put it anywhere around your house.

magpi.cc/smartipi | £30 / $25

▲ Official Raspberry Pi 4 Case

Simple and clean

This sleek and simple case is the official case for Raspberry Pi, acting as a perfect enclosure at a low price.

magpi.cc/pi4case | £5 / $5

▸ Gameboy NANO

Adorably tiny

One of the smallest cases we've seen for Raspberry Pi Zero, it also turns it into a handheld console. Great for small fingers.

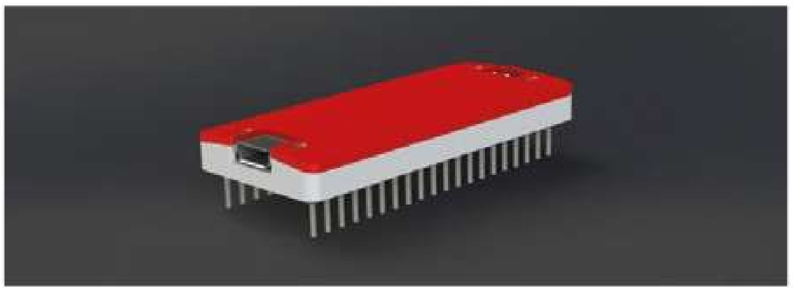
magpi.cc/gameboynano | Free

▸ Zebra Pico Case

Layered protection

Zebra makes lovely cases for microcomputers and dice, and this layered case for Pico offers great protection with full coverage for all the delicate parts, along with full access to GPIO and power.

magpi.cc/zebrapico | £8 / $9



▲ Raspberry Pi Pico Case

Officially inspired

While not an official case, this 3D-printed case is created in a similar style to the official ones, with a base and a lid.

magpi.cc/3dpicocase | Free

Official Raspberry Pi Zero Case

Simple and tiny

A companion case to the official case, this time shrunk down to fit all types of Raspberry Pi Zero.

magpi.cc/pizerocase | £5 / $5

Acrylic case for Raspberry Pi Pico

Two acrylic slabs

A very simple case made of two acrylic pieces that you join together through Pico's mount holes.

magpi.cc/acrylic | £8 / $10

FLIRC Raspberry Pi Case

For media PCs

This beautiful-looking case is made with aluminium, and acts as a heatsink. The design lets it sit neatly under a TV to power your media centre desires. There's also a Zero variant.

magpi.cc/flirc | £14 / $16

NESPi 4 Case

Retro console camo

This great classic console replica also allows you to use the front ports as you normally would, and add fake cartridges.

magpi.cc/nespi4 | £28 / $27

Pico USB Rubber Ducky Case

Converting enclosure

This 3D-printable case includes a nice little groove to put in a specific micro-USB to USB-A adapter, allowing you to easily plug a Pico into your computer of choice.

magpi.cc/duckycase | Free

Learn computer science with Raspberry Pi

Resources to help you get to grips with computer science on Raspberry Pi. By **Phil King**

Raspberry Pi Computer Courses

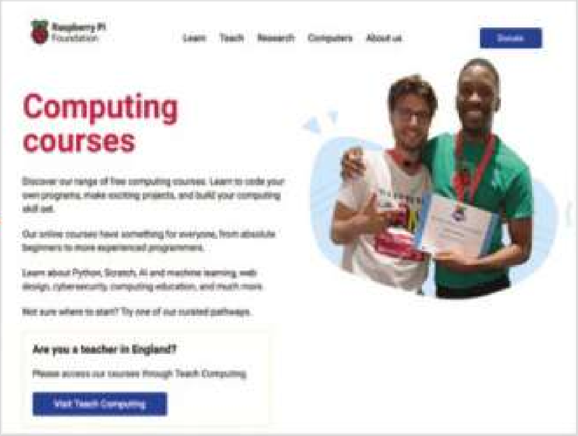
AUTHOR **Raspberry Pi Foundation**

Price: Free

magpi.cc/rpfcourses

There has never been a more important time to learn the fundamentals of computing, whether you have a related career in mind or you just want to feel more confident in how you deal with information technology. The Raspberry Pi Foundation was created with the aim of enabling every young (or older) person to develop that confidence while learning how to solve problems and express themselves creatively.

To that end, the registered charity has created a large range of learning resources aimed at various abilities and audiences. While the Raspberry Pi Projects site (**magpi.cc/projects**) offers numerous tutorials for computing and digital making, there's also an array of computer courses. The 'Computer Systems and Networking' category is a good place to start; among courses aimed at teachers, you can check out 'Understanding Computer Systems' and 'How Computers Work: Demystifying Computation'. Click on any course to enrol for free on FutureLearn.



Computing courses

Discover our range of free computing courses. Learn to code your own programs, make exciting projects, and build your computing skill set.

Our online courses have something for everyone, from absolute beginners to more experienced programmers.

Learn about Python, Scratch, AI and machine learning, web design, cybersecurity, computing education, and much more.

Not sure where to start? Try one of our curated pathways.

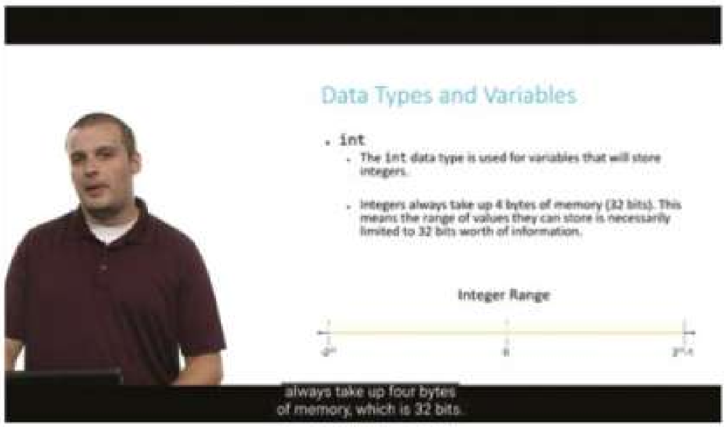
Are you a teacher in England?

Please access our courses through Teach Computing

Visit Teach Computing

Free courses

Learn computing with these free online courses



Data Types and Variables

- int
  - The int data type is used for variables that will store integers.
  - Integers always take up 4 bytes of memory (32 bits). This means the range of values they can store is necessarily limited to 32 bits worth of information.

always take up four bytes of memory, which is 32 bits.

CS50: INTRODUCTION TO COMPUTER SCIENCE

Harvard University's excellent entry-level course teaches you how to think algorithmically to solve problems. It covers a wide range of key topics and features a cloud-based IDE for coding tasks.

▶ magpi.cc/cs50

INTRODUCTION TO COMPUTER SCIENCE AND PROGRAMMING

While fairly old, this course from MIT gives beginners a good introduction to the subject and gets you writing and debugging Python programs. It includes video lectures, slides, and other resources.

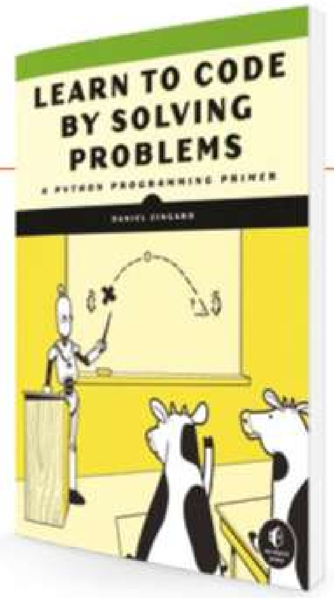
▶ magpi.cc/mitintrocs

PROGRAMMING 101

Comprising a series of video lectures, this Udemy course will provide you with a broad foundational knowledge of hardware, networking, programming, and licensing.

▶ magpi.cc/udemy101

Learn to Code by Solving Problems


LEARN TO CODE BY SOLVING PROBLEMS
A PYTHON PROGRAMMING PRIMER
DANIEL ZINGARO

AUTHOR
Daniel Zingaro

Price:
£30 / $35

magpi.cc/zingaropython

One of the fundamental aspects of computer science is learning how to code. For this, Python is one of the most popular and easy-to-learn languages, since it's well-structured and is used in all sorts of real-world applications.

Daniel Zingaro's programming primer is a suitable introduction for beginners, covering core concepts such as loops, lists, sets, functions, file management, sets, dictionaries, and algorithms. It concludes with a chapter on how to write more efficient code.

In addition, and perhaps more importantly, it teaches you the type of thinking required to solve problems using a computer. The key is to use top-down design to break a larger problem down into smaller tasks that can be tackled using functions. You can test your problem-solving skills on a series of exercises at the end of the book.

The Official Raspberry Pi Beginner's Guide

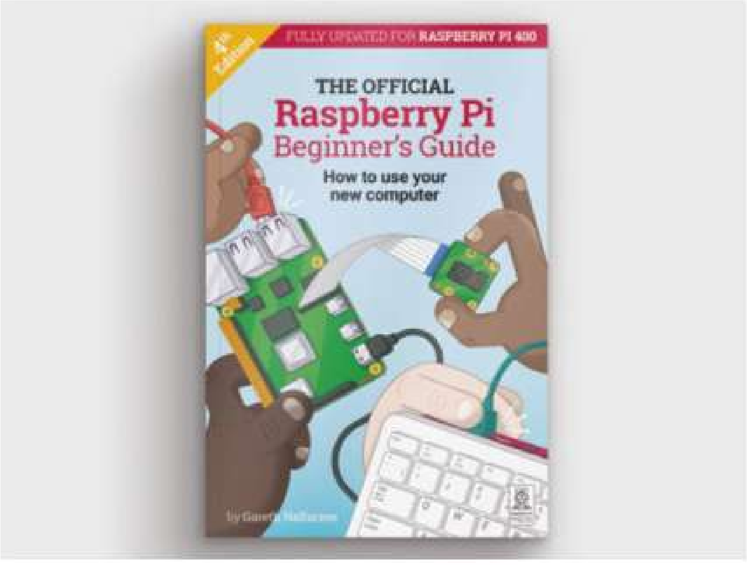
AUTHOR
Gareth Halfacree

Price:
£10 / $12 (Free PDF)

magpi.cc/bguide4

A Raspberry Pi computer represents an inexpensive and easy way to get into computing and digital making. This official guide book (also available as a free PDF, and in several languages) starts off by detailing how to set up your Raspberry Pi hardware and navigate its desktop operating system.

It then embarks on an easy-to-follow introduction to the fundamentals of programming – in chapters for both the block-based Scratch 3 and text-based Python languages – involving the creation of a variety of software projects and games. Not only that, but it then moves on to cover 'physical computing', controlling electronic circuits connected to Raspberry Pi's GPIO pins, via a number of fun projects. In short, you'll discover how computers work, learn to code, and start building some amazing things.



Books to read

Recommended reading for learning computer science

CODE: THE HIDDEN LANGUAGE OF COMPUTER HARDWARE AND SOFTWARE

Recently updated to an expanded 2nd Edition, Charles Petzold's book explores in detail how a computer works, gate by gate. There's a companion website at **codehiddenlanguage.com**.

▶ magpi.cc/codehidden

GROKKING ALGORITHMS

Algorithms are the building blocks of all computer programs. This book shows you how to apply common algorithms to the sorts of practical problems you may face as a programmer.

▶ magpi.cc/grokalgo

PYTHON CRASH COURSE

One of the easiest-to-follow programming books for beginners, it's project-based, so you'll be creating Python programs as you learn key skills that are applicable to most other languages.

▶ magpi.cc/pythoncrash

The Centre for Computing History interview

We talk about the incredible museum located in Cambridge, full of classic computers, retro games, and a giant processor

URL computinghistory.org.uk

Cambridge has many attractions for tourists. It's a very old city with a rich history and heritage, especially when it comes to computers – a lot were designed there, including Raspberry Pi. A trip to the Grand Arcade's Raspberry Pi Store should be on every tech fan's bucket list when in Cambridge, but you should also make sure to check out the Centre of Computing History.

It's regularly ranked highly on Cambridge 'off the beaten path' articles for visitors and for good reason; it's a wonderland of electronics from the dawn of computing to now, showing the evolution and ever-changing shape of PCs, mainframes, mobile phones, games consoles, and more.

"So it goes back to [August 2013]," Jason Fitzpatrick, the CEO and Curator, tells us. "It was originally my collection. My interest. My collection that was going a bit haywire.

"Business was good at the time, and I spent a bit of money on machines. It's the standard story any collector will recognise; you buy the machine that you had as a kid. Then you buy the machines you wanted as a kid, then machines that your friends had that you couldn't have, and then you find any reason to expand.

"I had machines like the BBC Micro and the Spectrum, up on shelves in the office at work. And it didn't matter who came in, you'd end up wasting the first 20 minutes on 'Oh, I had one of them.' It was just was obvious that, whilst I had a passion for it, everybody has something to say about it.

The history of portable computers makes it seem wild that laptops exist at all

Just about every major console is represented in the museum

> We built the website for it and, at that point, people started donating

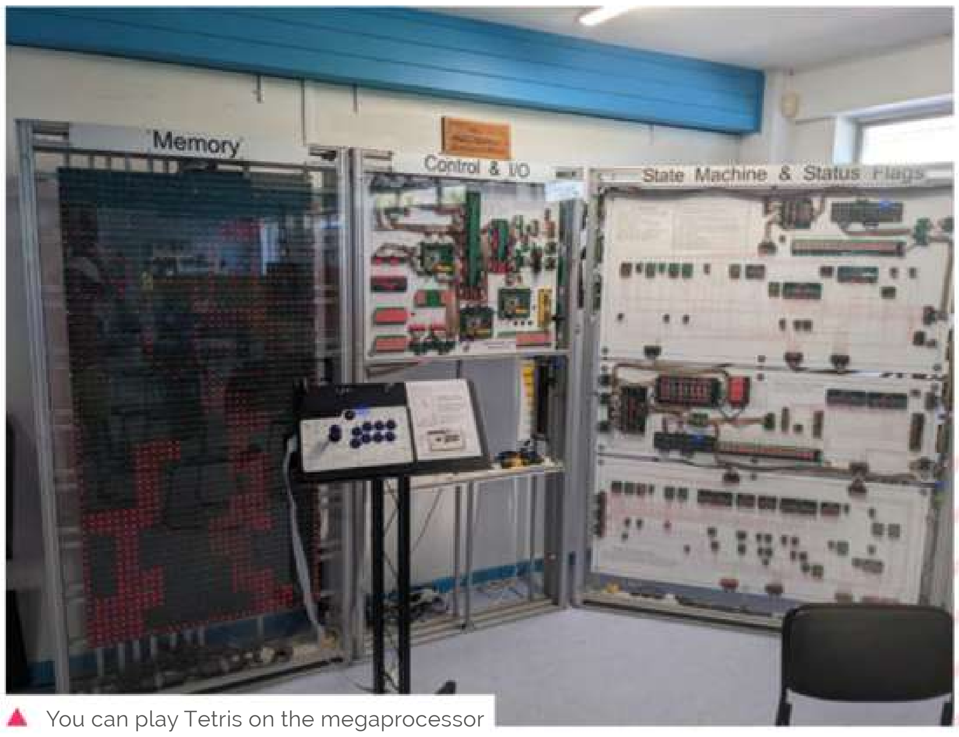
"[A group of us] got together and said, right, let's try and do something and create a museum. That museum was about half an hour away from Cambridge. We moved my company to a place where I could give it the ground floor. And the museum physically existed on this ground floor, but it was really small. We built the website for it and, at that point, people started donating. There were three small rooms and they all just got filled up within... it felt like weeks, but it was within a year. You couldn't move for donations and there was no display space left. So we closed again and it went in limbo a little bit for some time."

Due to a fortunate Twitter encounter, the team were able to be introduced to enthusiasts that allowed them to find a place in Cambridge, where the Centre for Computing History now lives.

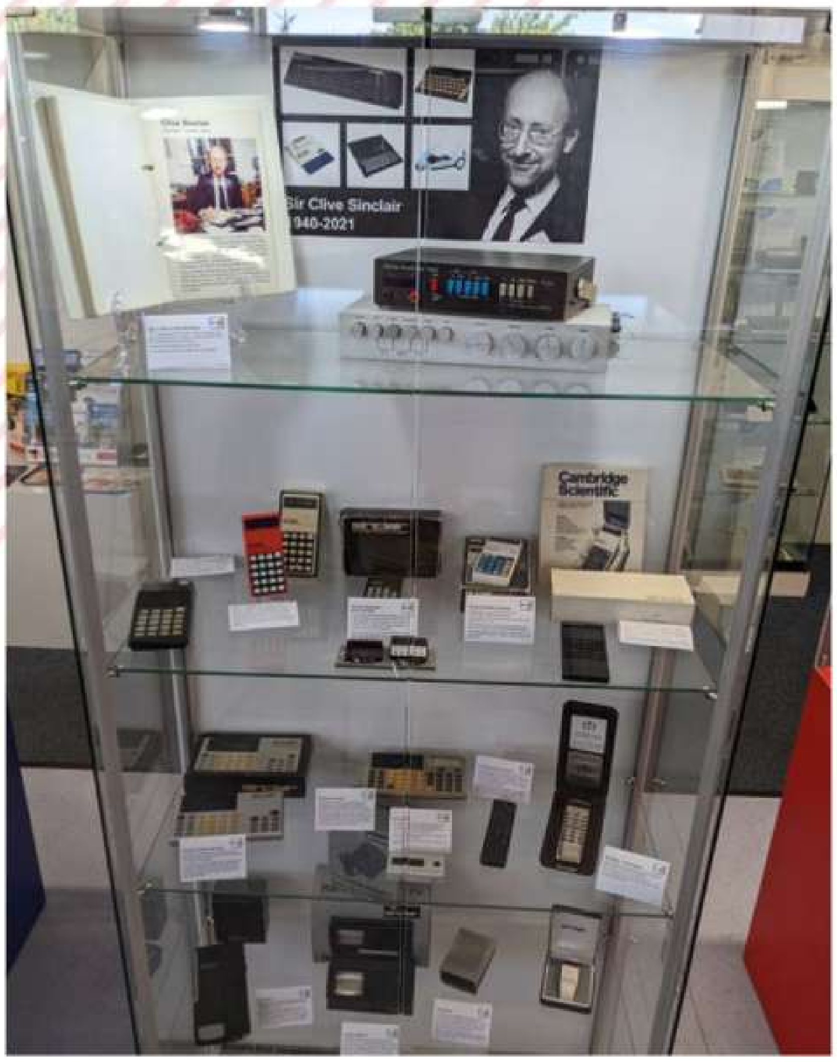
Building up

The museum now is crammed with computers and displays from many eras of many different kinds. An entrance area and canteen includes some arcade machines, some small displays that are changed out regularly, and the megaprocessor, an incredible recreation of a CPU from scratch using segmented displays that can be used to teach how a processor works. There's a classroom area with BBC Micros, Acorn computers, and Raspberry Pi 400s, and the main exhibit is in a large warehouse in the rear.

It wasn't always that way though – when it opened, computers were set out on tables with white cloths and not everything was painted yet.



You can play Tetris on the megaprocessor



After the passing of Sir Clive Sinclair, a display in his honour was set up

Acorns and Raspberry Pi 400 go hand in hand

However, as the collection grew, it became too big for it all to be displayed. There are roughly 160 pieces on display now, with over a thousand in storage. How do you pick what to display though?

"You've got to represent kind of the story of the time, although some things stay as the are," he explains. "We're really lucky as a computer museum, you can pitch your tent anywhere in terms of, if something happens in the news or whatever, you can usually do some sort of display around [it]. What haven't computers been involved with? Sadly, Sir Clive Sinclair passed away not long ago, so we have a little display of his stuff [in the entrance].

"So, we change things around and you can reflect popular culture and all sorts, so it's a bit fluid. We change things around a lot to keep it interesting for us, but also for visitors, because we do get a lot of repeat visitors. So it's nice when they can come back and see different things."

> We change things around and you can reflect popular culture and all sorts

Communities and schools

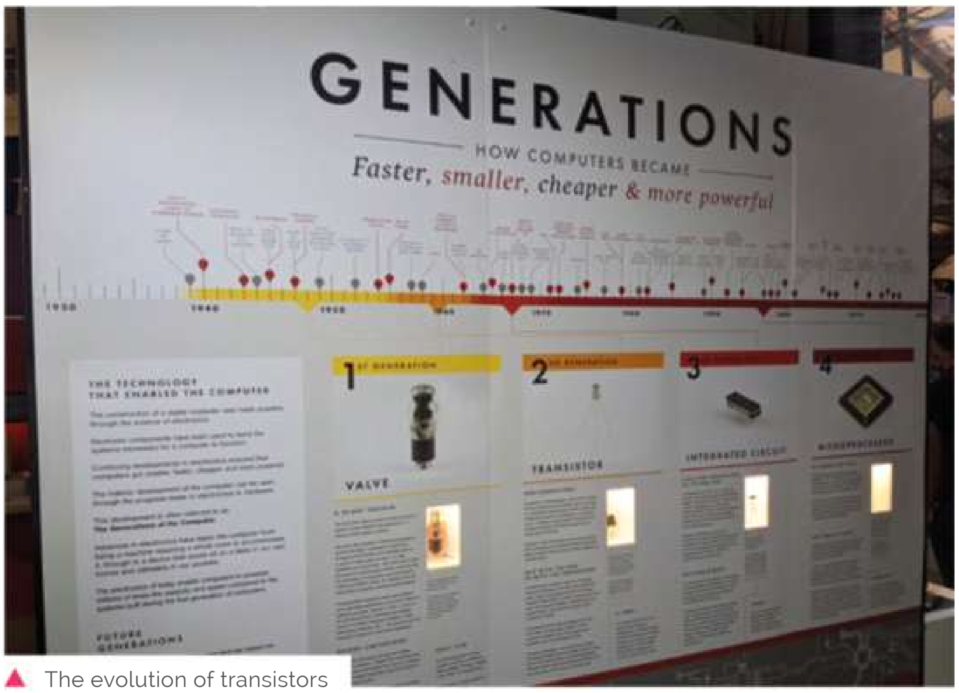
The museum also acts as a space for various computing and retro gaming communities to gather for an evening. Whether you're an Acorn Head or a C64 Fan, there's usually an event planned.

It's also often used by schools for a day trip during term time,

The main gallery is huge, and packed with amazing exhibits

There are many machines on display

Seeing the miniaturising of tech into one pocket device really puts it into perspective



The evolution of transistors

Not included: episodes of Bagpuss

with groups learning computing appropriate to their level through various activities.

"The feedback we get from schools is embarrassingly high," he admits. "It looks faked. The exit poll is 98%, nearly 99% positive. It's really cool. When the schools visit, they're super-busy and they don't get a great deal of time to just mess around. [Chris Monk, learning advisor], who designed the workshop and the system for teaching [allows us to] have 60 kids, so a lot to deal with. We split them into groups of 15; one lot in [the entrance area] working with the mega-processor on whatever level that suits. Then another lot in the classroom, doing whatever workshop they want – it might be electronics, and it might be programming, it might be TWINE, [it] can be BBC Micro. We still teach Basic on the BBC Micro because Basic and Python aren't that dissimilar. We'll have a tour going on in the main gallery, then the fourth group has got a bit of time to themselves."

While we visited, there were plenty of families around with their kids, checking out all the amazing displays and playing games from the 1970s and the 2000s, so it's not all educational. There's also several Raspberry Pi behind the scenes powering the fun.

MagPi Monday

Amazing projects direct from our Twitter!

Every Monday we ask the question: have you made something with a Raspberry Pi over the weekend? Every Monday, our followers send us amazing photos and videos of the things they've made.

Here's a selection of some of the awesome things we got sent this month – and remember to follow along at the hashtag #MagPiMonday!

01. Kevin's dog Archie looks amazing in his special LED coat

02. You can't see from this screen, but the spyglass is built into a Pringles can

03. Glad to see the Talkative Tube dashboard project from issue 120 is being hung nicely on the wall now

04. This higher tech, yet lower fidelity, version of Battleships looks very fun

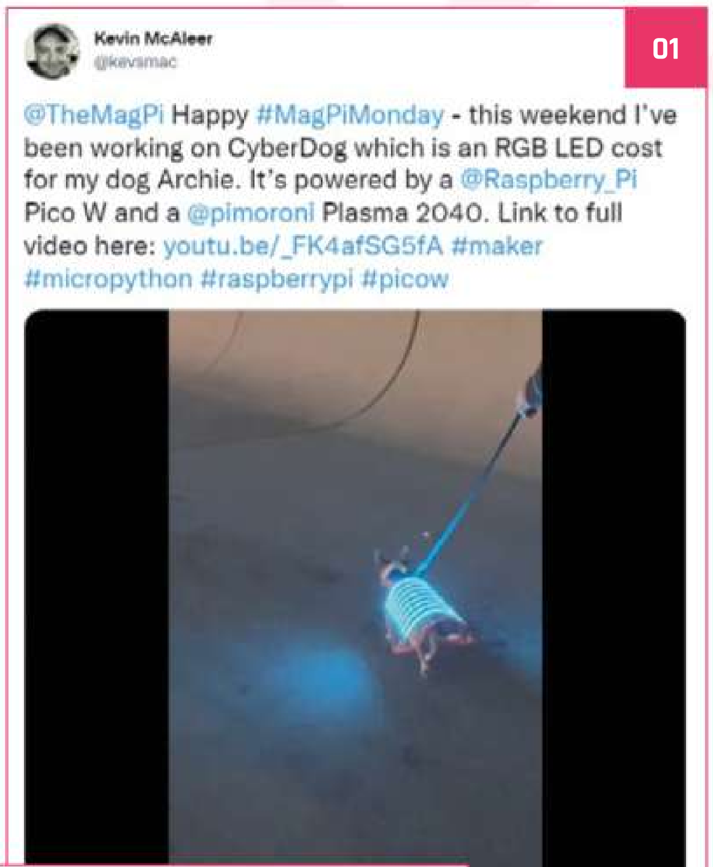
05. Be careful when gutting something like this – although the outcome is well worth it

06. We bet, at a slow speed, this would quite a nice and calming lamp for sleep

07. This is extremely pretty in motion!

08. Love the idea of this – there's something very cool about analogue meters

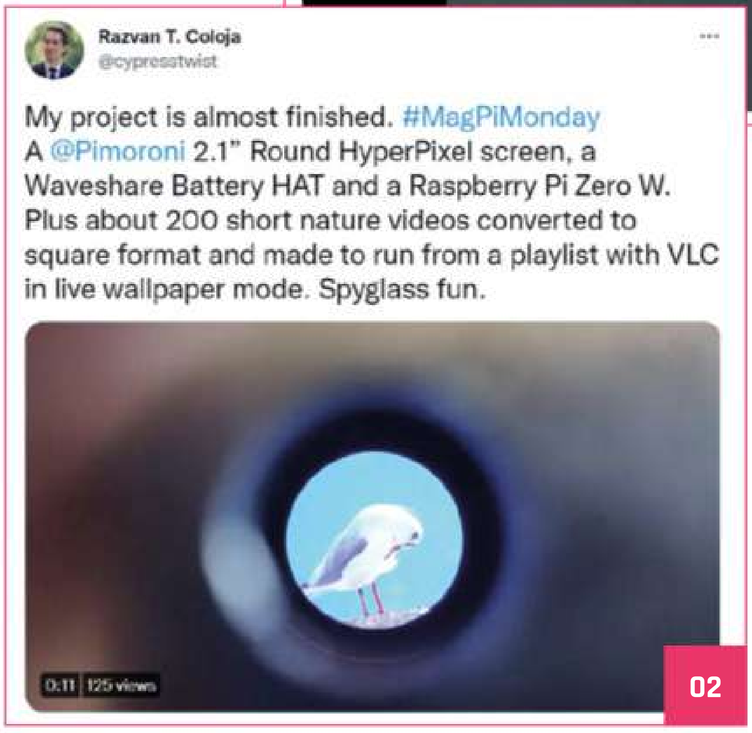
09. A nicely built looking camera! The Raspberry Pi High Quality camera can pull off some great shots



01

Kevin McAleer
@kevsmac

@TheMagPi Happy #MagPiMonday - this weekend I've been working on CyberDog which is an RGB LED cost for my dog Archie. It's powered by a @Raspberry_Pi Pico W and a @pimoroni Plasma 2040. Link to full video here: youtu.be/_FK4afSG5fA #maker #micropython #raspberrypi #picow

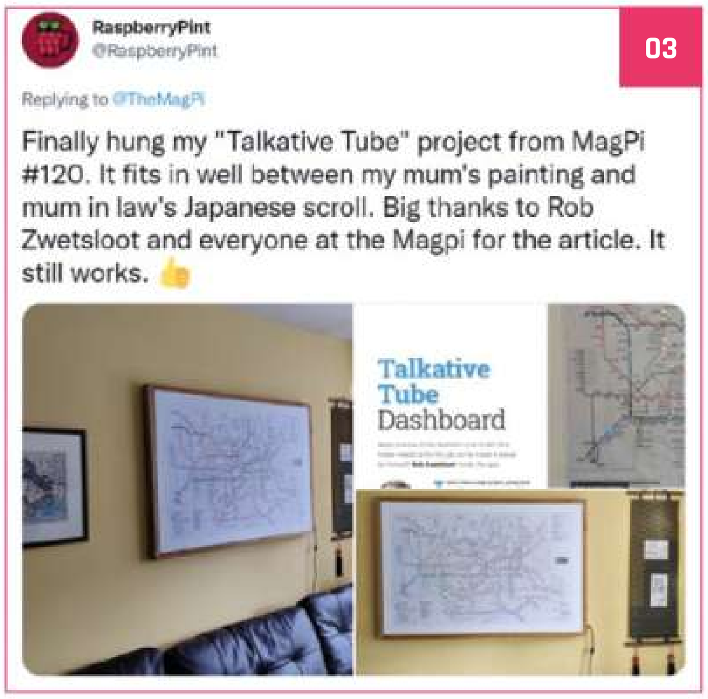


Razvan T. Coloja
@cypresstwist

My project is almost finished. #MagPiMonday
A @Pimoroni 2.1" Round HyperPixel screen, a Waveshare Battery HAT and a Raspberry Pi Zero W. Plus about 200 short nature videos converted to square format and made to run from a playlist with VLC in live wallpaper mode. Spyglass fun.

02

03

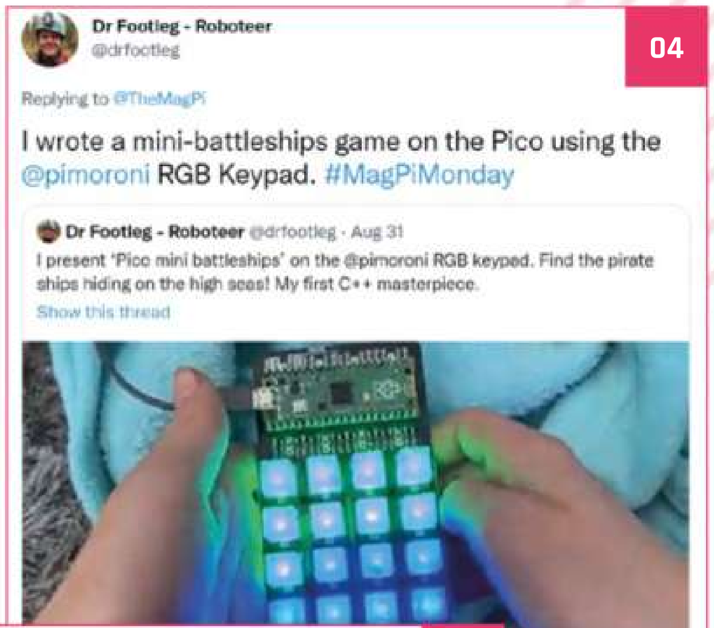


RaspberryPint
@RaspberryPint

Replying to @TheMagPi

Finally hung my "Talkative Tube" project from MagPi #120. It fits in well between my mum's painting and mum in law's Japanese scroll. Big thanks to Rob Zwetsloot and everyone at the Magpi for the article. It still works. 👍

04



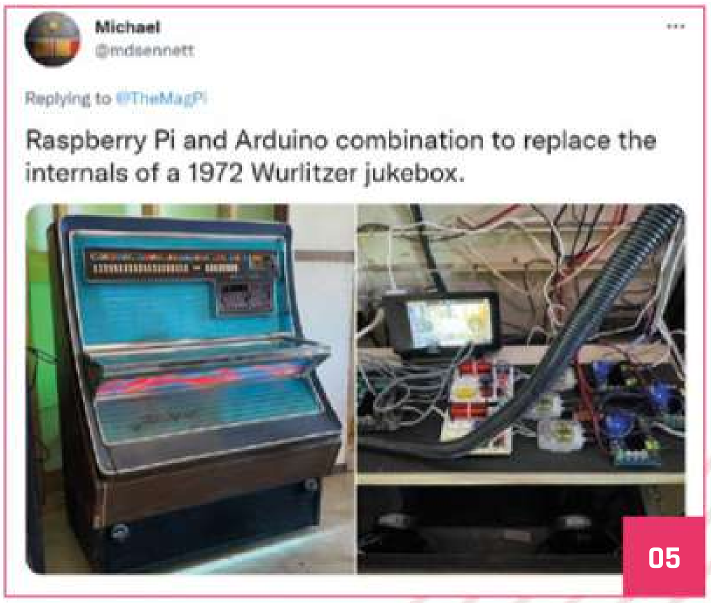
Dr Footleg - Roboteer
@drfootleg

Replying to @TheMagPi

I wrote a mini-battleships game on the Pico using the @pimoroni RGB Keypad. #MagPiMonday

Dr Footleg - Roboteer @drfootleg · Aug 31
I present 'Pico mini battleships' on the @pimoroni RGB keypad. Find the pirate ships hiding on the high seas! My first C++ masterpiece.
Show this thread

05

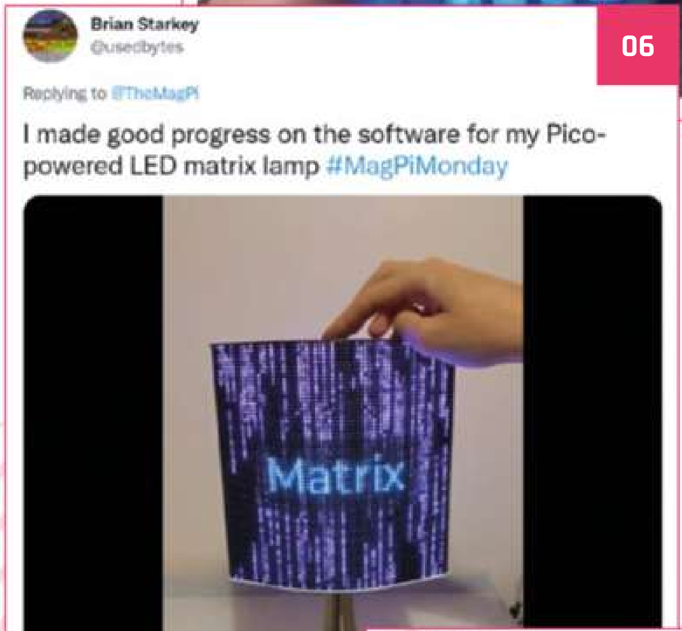


Michael
@mdsennett

Replying to @TheMagPi

Raspberry Pi and Arduino combination to replace the internals of a 1972 Wurlitzer jukebox.

06

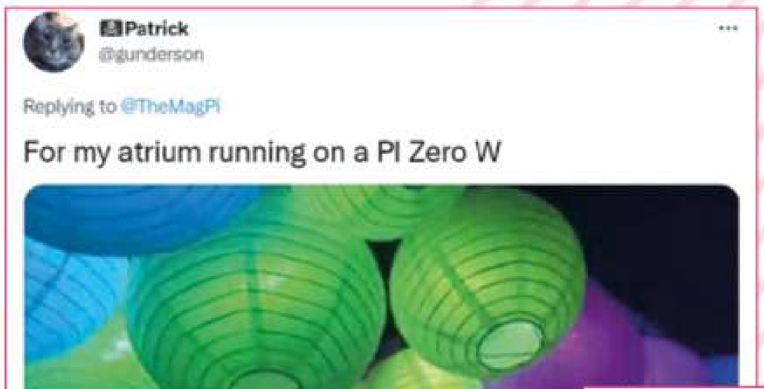


Brian Starkey
@usedbytes

Replying to @TheMagPi

I made good progress on the software for my Pico-powered LED matrix lamp #MagPiMonday

07



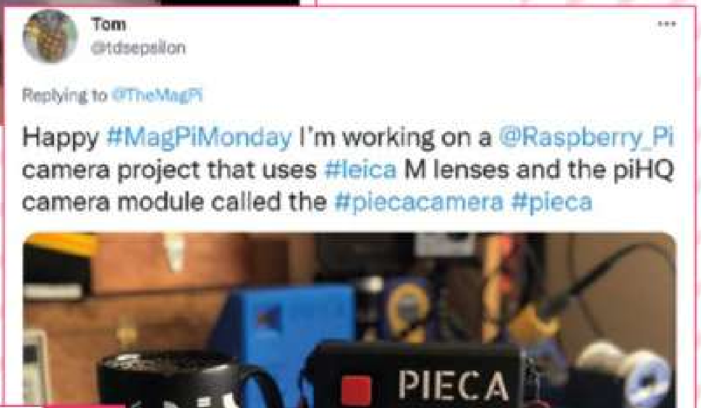
Patrick
@gunderson

Replying to @TheMagPi

For my atrium running on a PI Zero W

1,444 views

09

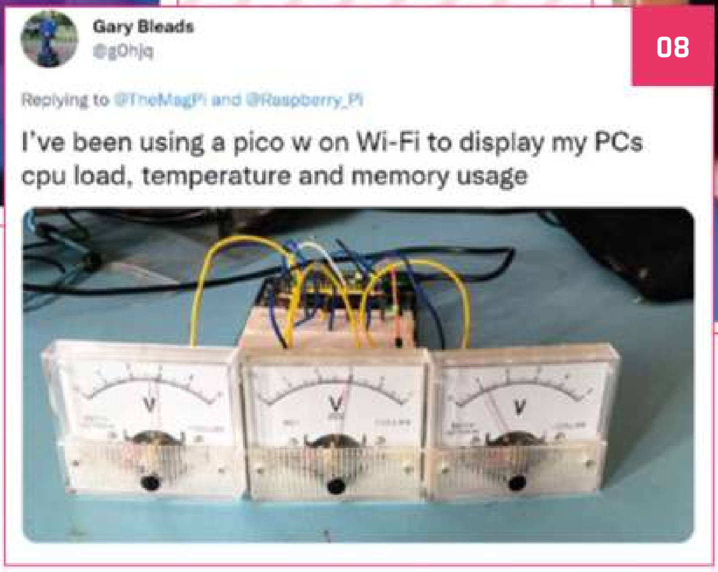


Tom
@tdsepsilon

Replying to @TheMagPi

Happy #MagPiMonday I'm working on a @Raspberry_Pi camera project that uses #leica M lenses and the piHQ camera module called the #piecacamera #pieca

08



Gary Bleads
@g0hjq

Replying to @TheMagPi and @Raspberry_Pi

I've been using a pico w on Wi-Fi to display my PCs cpu load, temperature and memory usage

Raspberry Pi at SiliCon!

The cosplay and maker convention was a hubbub of cool costumes and cooler tech

As restrictions are lifting, conventions are coming back, and that means you'll start seeing Raspberry Pi booths popping up at tech- and maker-related cons. The team already attended Embedded World earlier in the year and, this time, they were in attendance at SiliCon in San Jose, California. It's a cosplay and maker con from Adam Savage of Mythbuster's fame, and the team was there to show off just how good Raspberry Pi Pico could be for making and cosplaying.

Features Ed Rob went along to help out and show off Pico, as well with the Poltergust G-00 that you can build on page 58. It was great to get back to these kind of big cons, with a bustling artists alley full of excellent pieces, weird and wonderful crafts for sale from all corners of the makersphere, and just a lot of nerd stuff on display to appreciate.

We spoke to Matt Richardson in *The MagPi* 120 (**magpi.cc/120**) about joining Raspberry Pi Ltd from the Foundation to work on events, and one of the things he told us was about the excitement of people meeting someone from Raspberry Pi.

"It's funny when we're at an event and someone comes up to us, completely taken aback by the fact that they're seeing someone from Raspberry Pi in-person," Matt said. "Many people still think of us as a tiny group of people who make an affordable computer. They really don't expect us to be out in the world sharing the 'good word' about Raspberry Pi."

There were plenty of folks coming up to chat about stuff they'd made, stuff they liked, and more. Hopefully Rob will get to see you at a future event with Raspberry Pi.

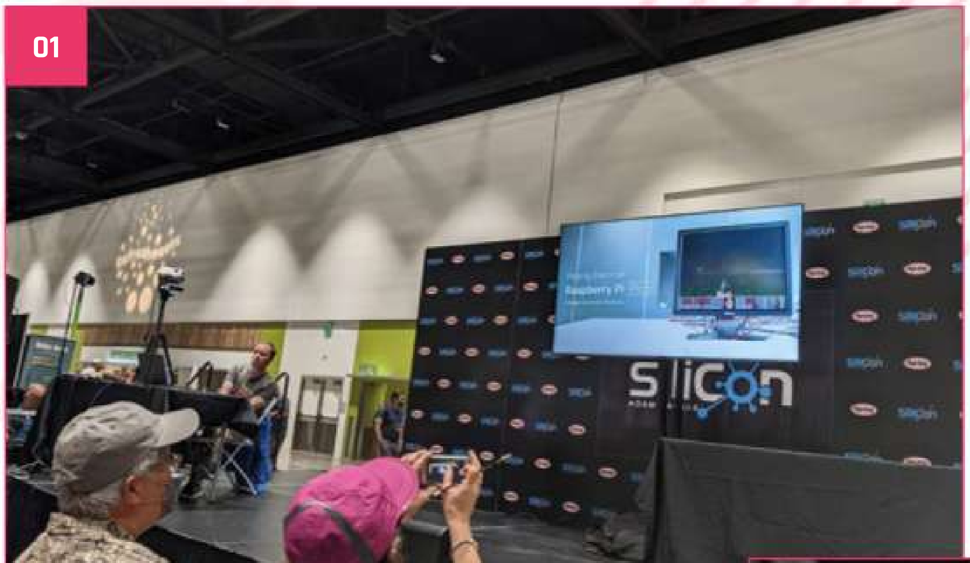
01. Alasdair Allan shows off how easy it is to program LEDs on Pico, and also how you can run DOOM on it

02. We saw a few excellent robots at the event, including this wonderful shoulder-mounted one

03. Tim Stenning, from the Raspberry Pi Store, was on site to help folks with learning the ropes with Pico

04. Matt had sorted out a fun demo for people to try out and learn about Pico

05. Free swag and many Pico projects were on display for folks to look at

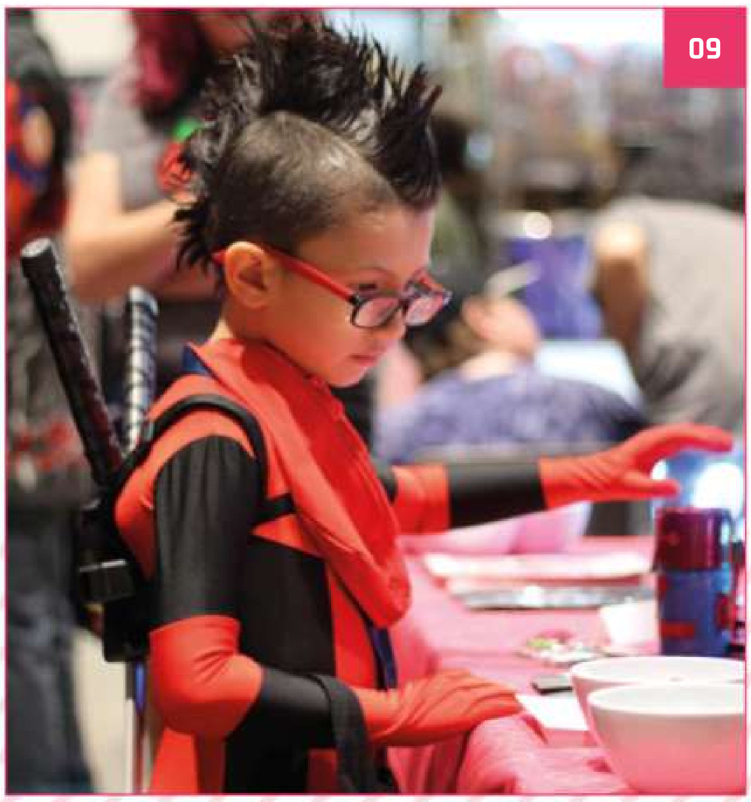

01

02

03

04

05



06. The whole booth was quite simple, but showed off the strengths of Pico

07. The Cosplay Grade RX-78-2 Gundam – complete with accurate decals from the model kit – was keeping the convention safe

08. Features Ed Rob helps out someone with their Pico code

09. A young *Deadpool* cosplayer is mysteriously drawn to the *Spider-Man* robot

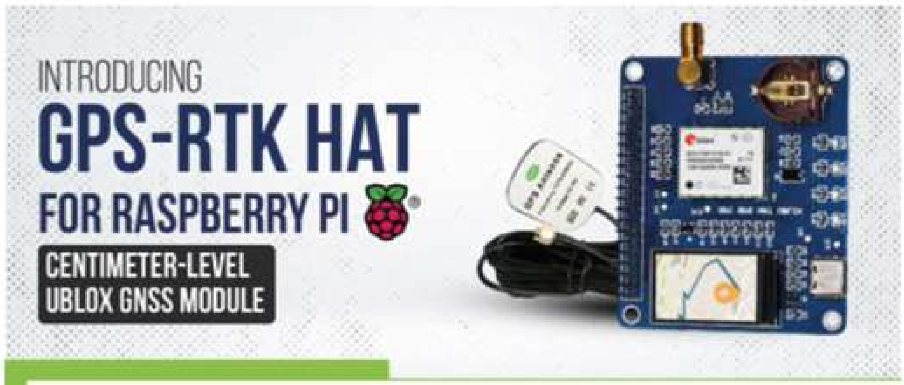
Crowdfund this!

Raspberry Pi projects you can crowdfund this month

The Red Reactor

"The Red Reactor is a high-performance battery power board designed to meet high-level requirements, and has been tested cleanly delivering in excess of 4 amps to a wide range of designs with the Raspberry Pi 4, Model 3 (A+ and B+), and Pi Zero, along with ancillaries such as a display, USB hub for keyboard, mouse, and webcam, and even an audio amplifier."

▶ kck.st/3DqT9Kj



GPS-RTK HAT

"This ZED-F9P-based Raspberry Pi GNSS HAT provides centimetre-level accuracy in seconds. It offers features such as: multi-band RTK with quick convergence times, a high update rate, support for moving base RTK mode, simultaneous reception of four GNSS systems, support for augment positioning systems, accurate and quick positioning with minimal drifting, and anti-spoofing and anti-jamming capabilities."

▶ kck.st/3RvM8fm

Your Letters

Saving energy

I have an idea for an article or an article series. Energy consumption should be at the lowest possible in climate crisis times, so maybe you could run programs on low-cost Raspberry Pi that help monitor and save energy.

Jürgen via email

That's a great idea. We are doing a feature on using Pico for smart appliances and home monitoring in an upcoming issue. We will be sure to include an energy-saving aspect to it. In the meantime, be sure to look at using Energenie products with Raspberry Pi to control and monitor electrical usage (**magpi.cc/energenie**).

Monitoring and saving energy is an important part of the modern world

Archie the CyberDog on a flashy walkabout

Rainbow dog

Happy #MagPiMonday (magpi.cc/magpimonday). I've been working on CyberDog, which is an RGB LED coat for my dog Archie. It's powered by a Pimoroni Plasma 2040. There is a link to full video here: magpi.cc/cyberdog.

Kevin via Twitter

That's an excellent build Kevin, and Archie looks amazing in that rainbow coat. We will be getting in touch and doing an interview with you on that build. Other readers, be sure to share your projects with us on Twitter using the hashtag #MagPiMonday.

Contact us!

Twitter	@TheMagPi
Facebook	magpi.cc/facebook
Email	magpi@raspberrypi.com
Online	forums.raspberrypi.com

Great balls of fire

I just wanted to let you know how much I enjoyed reading about Fireball Aotearoa [see *The MagPi* issue 121, magpi.cc/121 - Ed]. I'm an amateur astrophotographer, and it never occurred to me that you could use the power of Raspberry Pi with video cameras to capture and locate falling meteorites. Whatever will people think of next? Bravo to Scott, Taylor, and Rowe.

.

Thomas via email

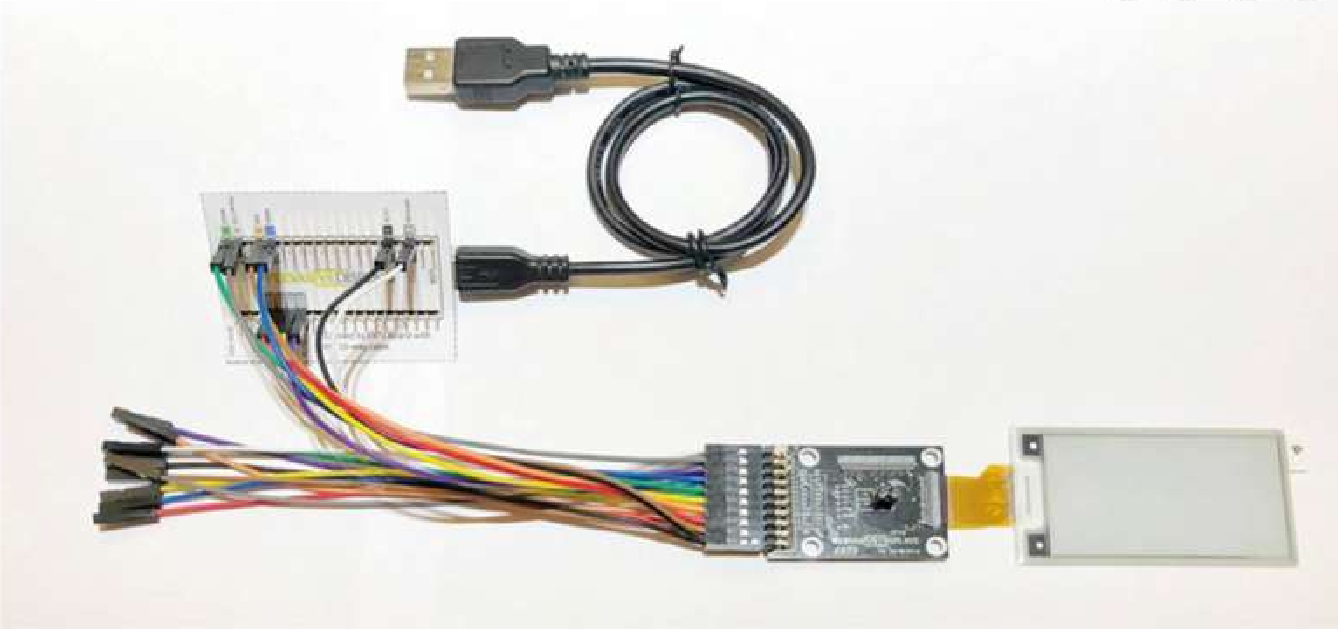
Yes. It is a fantastic use of Raspberry Pi, and we hear the team is expanding the project to capture even more falling meteors. We are glad you enjoyed it, and be sure to keep reading into the winter months as we have more night sky features then.

▲ This camera locks on to a fireball as it travels across the sky and reports the meteor landing location

WIN 1 OF 5

EPD PICO KITS

This kit allows you to add an e-ink display to Pico for extremely pretty looking projects that don't need to update their info often. We reviewed it last issue and thought it was quite powerful, with a lot of great functions. Now is your chance to get one!



Head here to enter: magpi.cc/win | **Learn more:** magpi.cc/epdk

Terms & Conditions

Competition opens on **28 September 2022** and closes on **27 October 2022**. Prize is offered to participants worldwide aged 13 or over, except employees of Raspberry Pi Ltd, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed or administered by, or associated with, Instagram, Facebook, Twitter or any other companies used to promote the service.

BUILD YOUR PICO W SMART HOME

INSTALL INTERNET-CONNECTED SMART DEVICES AROUND THE HOUSE

THE MAGPI **#123**
ON SALE **27 OCT**

Plus!

Build a Pico Plant Monitor

Learn computer science skills

Control sounds with Pico sequencer

DON'T MISS OUT!
magpi.cc/subscribe

TWITTER	@TheMagPi
FACEBOOK	fb.com/MagPiMagazine
EMAIL	magpi@raspberrypi.com

EDITORIAL
Editor
Lucy Hattersley
lucy@raspberrypi.com
Features Editor
Rob Zwetsloot
rob@raspberrypi.com
Sub Editor
Nicola King

ADVERTISING
Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN
criticalmedia.co.uk
Head of Design
Lee Allen
Designers
Olivia Mitchell, Ty Logan,
Sam Ribbits,
Illustrator
Sam Alder

CONTRIBUTORS
David Crookes, PJ Evans,
Rosemary Hattersley, Nicola
King, Phil King, Simon Long,
Sean McManus, KG Orphanides

PUBLISHING
Publishing Director
Russell Barnes
russell@raspberrypi.com

Director of Communications
Liz Upton
CEO
Eben Upton

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS
Unit 6 The Enterprise Centre
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE
+44 (0)1293 312193
magpi.cc/subscribe
magpi@subscriptionhelpline.co.uk

Retro education

How retro computers, not typist courses, made all the difference

Our big theme in this edition of *The MagPi* has been retro computing. I remember getting into computers when I was little. One day, a BBC Micro turned up at school and I was hooked. My dad took some persuading, but eventually I sweet-talked him into buying the family a ZX Spectrum 48K. Neither of my two older sisters was interested, so I got it all to myself.

I loved the way you could create, and see, things on the screen, how programs could be used to make things. While other people saw glorified calculators, I was entranced with interactive stories like *The Hobbit* and *The NeverEnding Story*. When I wasn't reading fiction, I was lost in a world of pixellated narrative.

When I wasn't reading books or playing adventure games, I was learning BASIC and making my own creations. Computing classes covered everything from acid-etched circuits to PASCAL programming.

As I worked my way through school, the computer became less about exploring the potential of a wonderful new gizmo and more about fitting students into the mould of office life. By the time I went to college, I was being taught touch-typing and how to format word-processing documents.

My love of narrative was filled with English and art, and computing became a personal hobby and less of an academic study. This is a shame because computers are incredible things. Packed with unlocked potential that can be used in all kinds of scientific and creative endeavours. Without computers, we wouldn't have modern art in its current form; or video, film, or modern music. Still, I managed to bring it all together by editing a magazine about computers and taking courses at MITx.

> “ Cutting-edge technology can trace a path back to earlier days ”

It's no surprise to me that some of the most creative people I've ever met are quite nerdy; and vice versa. The secretarial training was the opposite of both: learning by rote with no understanding. It was bland. In retrospect, I should have taken the electronic typewriter apart to see what was going on inside.

I like to think that Raspberry Pi and the new Pico microcontrollers can fill the tech void that is teaching, otherwise smart kids, basic office skills to tick government check boxes. Simply putting bare tech in front of kids who are interested can make all the difference.

Our retro gaming with Pico feature obviously appeals to retro geeks like me who remember the 8-bit days with love and affection. I learnt coding and computing by experimentation. Retro computing on a modern device, like Pico, also shows how cutting-edge technology can trace a path back to earlier days and help us gain an understanding of computing development.

AUTHOR

Lucy Hattersley

Lucy is editor of *The MagPi* and is dancing like nobody is watching. Everybody is watching.

magpi.cc

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiShop.us PiShop.ca PiShop.mx ThePiHut Pi-Shop.ch Welectron.

Contact your favorite Pi store if it's not listed here

HiPi.io

PiKVM

Manage your servers or
workstations remotely

A **cost-effective** solution for data-centers,
IT departments or remote machines!

PiKVM HAT
for DIY and custom projects

Pre-Assembled version

- Real-time clock with rechargeable super capacitor
- OLED Display
- Bootable virtual CD-ROM & flash drive
- Serial console
- Open-source API & integration
- Open-source software

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz Váš dodavatel Raspberry Pi · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · PiShop.mx

Reseller suggestions and inquiries:
wholesale@hipi.io